PARALLEL UNMASKING UNDER UNKNOWN DEPENDENCE: EXACT MINIMAX BATCHING

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion language models decode multiple tokens in parallel to accelerate generation, but tokens in the same batch cannot use one another's values, so parallel decoding may fail to preserve their dependencies. Existing decoders use confidence or estimated dependence to set batch sizes, yet the best worst-case guarantee remains unknown when batch sizes must be chosen without reliable dependency information. We exactly characterize this limit for every sequence length and call budget, construct a plan that attains it, and give matching targets showing no fixed deterministic plan can do better when vocabulary is large enough. The key is to track when dependence becomes relevant during decoding. Each plan can be judged by its risk at every stage, and the optimum minimizes the largest. We also quantify the gains from randomization when the chosen schedule is known, identify when online adaptation cannot improve the boundary, and exactly separate batching from predictor error. On LLaDA and Dream, theory-derived batch sizes improve generation accuracy over equal batching with the same position-selection rule and call budget.

1 INTRODUCTION

Diffusion language models generate text by filling several masked positions in one model call, reducing the number of calls needed to complete a sequence (Austin et al., 2021; Lou et al., 2024; Sahoo et al., 2024). Tokens filled together see the same pre-call context and cannot condition on values produced alongside them. When those tokens depend on one another, the resulting joint output may fail to preserve that dependence. Decoding fewer tokens at a time restores more sequential conditioning but gives up parallel speed (Luxembourg et al., 2026). Every parallel decoder must therefore decide both which positions to fill and how many to fill in each call. When reliable dependency information is unavailable, for a sequence of n tokens, how should a budget of K calls be divided across rounds, and what is the best guarantee any batch-size rule can provide?

Existing work uses increasingly rich information to choose positions and batch sizes. Profile-aware theory computes target-specific schedules from supplied or estimated dependence curves (Chen et al., 2026; Lavenant & Zanella, 2025; Wainwright, 2026). Practical decoders use confidence, entropy, learned orders, conditional-independence tests, or dependency estimates (Jazbec et al., 2025; Xu et al., 2026; Ben-Hamu et al., 2025; Azangulov et al., 2025; Ringel et al., 2026; Kim et al., 2026; Luo et al., 2026b; Zhang et al., 2026; Luo et al., 2026a; Mohamed et al., 2025), while other methods modify the model or generation process (Chen et al., 2025a;b; Wang et al., 2025; Sun et al., 2026), sample tokens jointly, or allow revision and remasking (Shen et al., 2026; Bansal & Sanghavi, 2025; Wang et al., 2026; Dmitriev et al., 2026; Xiang et al., 2026; Jiang et al., 2025). Parallel-sampling theory and distribution-independent random schedules provide complementary limits (Li & Cai, 2025; Bansal et al., 2026; Zhao & Cai, 2026). These works provide informed schedules and universal guarantees, but leave open the exact finite limit for batch sizes chosen without reliable dependency information. Without this limit, gains from locating dependencies remain mixed with gains from allocating token counts across calls. With batch sizes fixed in advance, the same schedule must handle targets whose dependencies become relevant at different stages of decoding. Determining the exact limit therefore requires an error guarantee valid for every target, together with target distributions showing that no fixed schedule can guarantee less error.

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.

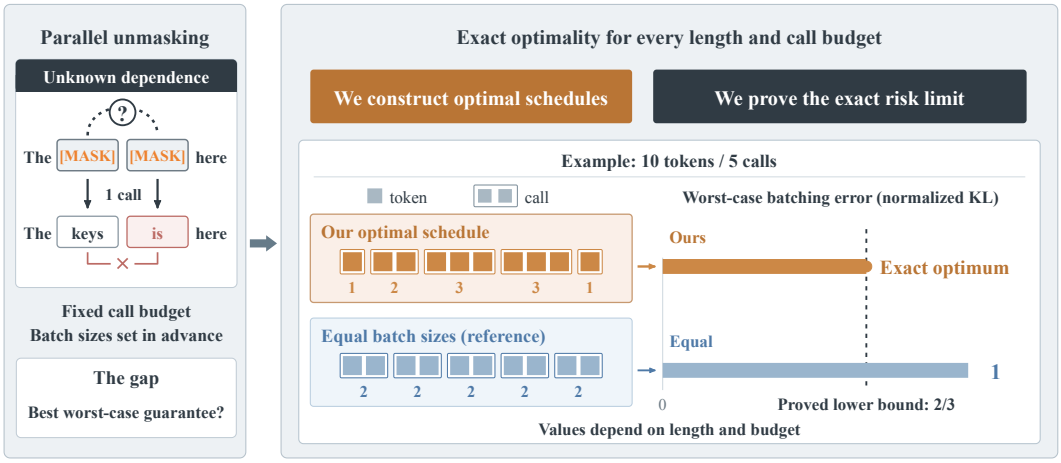

Figure 1: **Exact optimal batching under unknown dependence.** Left: tokens filled in one call can break a dependency ("keys" and "is"), yet batch sizes must be set before the dependence is known. Right: for ten tokens in five calls, our schedule attains worst-case normalized KL $2/3$, and under the conditions of Theorem 4.2 no deterministic five-call schedule does better (dashed line); equal batches reach 1. KL is normalized by the smaller of total and dual total correlation.

Our insight is to track when dependence becomes relevant as tokens are revealed one at a time. Each reveal may make additional dependence available to later tokens. A parallel batch skips some of these intermediate reveals, and its boundaries determine how much conditioning would be missed at each stage. A worst-case target can concentrate its dependence at the stage a plan handles least well, so the best plan minimizes the largest stage risk. This turns a problem over arbitrary joint distributions into a finite optimization over batch boundaries. It yields the exact finite limit, a plan that attains it, and matching target distributions that prove the limit cannot be improved.

Our contributions are threefold.

- **Exact limit under unknown dependence.** We exactly characterize the best worst-case guarantee for deterministic batch sizes under unknown dependence and give matching target distributions under the stated vocabulary condition. We construct an attaining plan in $O(K \log n)$ operations and prove that the guarantee composes across sequential blocks through the hardest block.
- **Limits of randomization and adaptation.** We solve the randomized problem when the sampled schedule remains observable and show that random batch sizes halve the leading worst-case loss in a broad parallel regime. In the exact-oracle, irrevocable, labelled setting, online batch-size adaptation and value-dependent position selection cannot improve the corresponding boundary, with the latter result holding at fixed cardinalities. When the schedule is hidden, we solve a continuum of fixed mixtures in the smallest case and bound its optimized binary value.
- **A connection to learned decoders.** We prove an exact decomposition that separates batching error from learned-predictor error. On held-out finite distributions, oracle output risk predicts schedule ordering under learned outputs. On LLaDA, theory-derived batch sizes improve generation accuracy over equal batching under the same position ranking and call budget. The improvement also appears on Dream under uniform position selection, with matched runtime.

2 Background and Motivation

Correct token marginals do not guarantee a correct joint output. Consider two masked tokens that must agree, while the current context makes either shared value equally likely. A perfect one-token predictor assigns the correct marginal distribution to each token. If the tokens are generated sequentially, the first value determines the second. If they are sampled independently in one call, they disagree half of the time. The error comes from removing the conditioning between the two

tokens, not from inaccurate one-token predictions. Batch size therefore matters even before model approximation enters the analysis.

Choosing the positions does not determine how many tokens to decode. One rule decides the order in which masked positions are considered. Another cuts that ordered list into parallel batches. Practical decoders often couple the two by using the same confidence or dependency estimate both to rank positions and to decide where a batch should end. The decisions nevertheless remain distinct because every chosen order still requires batch boundaries. Our theory isolates batch size under a uniformly random position order; the main LLM comparison fixes an existing decoder's order and changes only batch boundaries.

An information-free reference isolates what batch size contributes. Confidence and revealed values can identify safe batches when they expose the relevant dependence. They may also fail to distinguish safe and unsafe choices. Two tokens can each have high marginal confidence while their joint prediction is inconsistent, and a dependence estimate learned on one target may not transfer to another. A plan chosen before target information is available provides a reference for these cases. When reliable information is available, the same reference can be refined. Comparing against it separates the benefit of locating dependencies from the benefit of allocating token counts across calls.

The remaining gap is the exact limit when dependence cannot be located. This limit can be found without recovering the hidden dependency structure. Imagine revealing tokens one at a time in this random order. If new dependence becomes available after a reveal, later tokens could use it under sequential decoding, whereas a parallel batch commits them before it appears. Batch boundaries therefore assign a known exposure to every possible reveal stage. The target determines where dependence appears, and the batch sizes determine how much conditioning is lost there. Section 3 formalizes these two parts as dependence jumps and schedule loads. The exact limit is governed by the most exposed stage.

3 From unknown dependence to schedule loads

This section turns the schedule-before-dependence question into a finite load problem. Fix $n \geq 3$, a finite alphabet $\mathcal{X}$ of size q, and a law p for $X = (X_1, \ldots, X_n) \in \mathcal{X}^n$. We use a field alphabet only to construct the matching hard targets; all upper bounds below hold for every finite alphabet.

Schedules and two losses. A deterministic K-batch schedule is an increasing sequence

$$\mathcal{N} = (N_1, \ldots, N_K), \qquad 1 = N_1 < \cdots < N_K \leq n, \qquad N_{K+1} = n + 1. \tag{1}$$

Let $\mathfrak{N}_{n,K}$ denote the set of schedules satisfying Eq. (1). For a uniform permutation π, batch a comprises reveal positions $N_a, \ldots, N_{a+1} - 1$. The exact one-coordinate sampler has law

$$q_{\pi,\mathcal{N}}(x) = \prod_{a=1}^{K} \prod_{j=N_a}^{N_{a+1}-1} p\big(x_{\pi_j} \mid x_{\pi_1}, \ldots, x_{\pi_{N_a-1}}\big)\,. \tag{2}$$

We study two losses:

$$E_{\mathcal{N}}(p) = \mathbb{E}_\pi D_{\mathrm{KL}}(p \| q_{\pi,\mathcal{N}}), \qquad \widetilde{E}_{\mathcal{N}}(p) = D_{\mathrm{KL}}(p \,\|\, \mathbb{E}_\pi q_{\pi,\mathcal{N}})\,. \tag{3}$$

The first retains the sampled order while measuring KL and then averages; the second first mixes the sampler outputs and measures the law actually produced. Uniformity concerns the reveal order, not the target. For a fixed deterministic schedule the two minimax values coincide; with randomized cardinalities, the two losses require separate analyses.

Two dependence budgets. We normalize batching error by

$$\mathrm{TC}(p) = \sum_r H(X_r) - H(X), \tag{4}$$

$$\mathrm{DTC}(p) = H(X) - \sum_r H(X_r \mid X_{-r}). \tag{5}$$

These quantities measure total dependence from opposite reveal directions, so early and late dependence can create different obstructions for a schedule.

Information jumps and schedule loads. For uniformly chosen S of size $j-1$ and $i \notin S$, use the average information profile of Chen et al. (2026),

$$Z_j(p) = \mathbb{E}_{S,i} I_p(X_i; X_S), \tag{6}$$

where the expectation is joint over the uniform size-$(j-1)$ set and a uniform coordinate outside that set. Its adjacent jumps

$$\delta_i = Z_{i+1} - Z_i \geq 0 \tag{7}$$

record how much additional dependence appears after the ith reveal. If i lies in the interval $N_a \leq i < N_{a+1}$, set

$$d_{\mathcal{N}}(i) = N_{a+1} - i - 1, \qquad w_i = \min\{i, n-i\}. \tag{8}$$

The load $d_{\mathcal{N}}(i)$ counts the later positions in the same batch that cross the ith dependence jump.

Lemma 3.1 (Schedule-load identity)**.** The profile identities, written in jumps, give

$$E_{\mathcal{N}}(p) = \sum_{i=1}^{n-1} d_{\mathcal{N}}(i)\delta_i, \tag{9}$$

$$\mathrm{TC}(p) = \sum_{i=1}^{n-1} (n-i)\delta_i, \tag{10}$$

$$\mathrm{DTC}(p) = \sum_{i=1}^{n-1} i\delta_i. \tag{11}$$

Equivalently, the two dependence budgets satisfy

$$\mathrm{TC}(p) = \sum_{j=1}^{n} Z_j, \qquad \mathrm{DTC}(p) = nZ_n - \sum_{j=1}^{n} Z_j. \tag{12}$$

The identity exposes the interaction that matters: the target chooses nonnegative dependence jumps, and the schedule determines how often each jump is crossed within a batch. The unknown distribution therefore enters through one jump vector, while the schedule enters through one load vector.

The symmetric robust objective. For a fixed schedule, the sharp constants in the separate inequalities $E_{\mathcal{N}} \leq A_{\mathcal{N}}\,\mathrm{TC}$ and $E_{\mathcal{N}} \leq B_{\mathcal{N}}\,\mathrm{DTC}$ are

$$A_{\mathcal{N}} = \max_i \frac{d_{\mathcal{N}}(i)}{n-i}, \qquad B_{\mathcal{N}} = \max_i \frac{d_{\mathcal{N}}(i)}{i}. \tag{13}$$

When the dependence orientation is unknown, the smallest common scale-free coefficient is $\max\{A_{\mathcal{N}}, B_{\mathcal{N}}\} = \max_i d_{\mathcal{N}}(i)/w_i$. Geometrically, it is the first diagonal contact of the full TC/DTC coefficient region. If asymmetric costs are known, the corresponding point on the Pareto frontier in Appendix I is the appropriate objective; the diagonal is the symmetric robust choice.

The deterministic schedule-first value is therefore

$$R^{(q)}_{n,K} = \min_{\mathcal{N}} \sup_{p \neq \text{product}} \frac{E_{\mathcal{N}}(p)}{\min\{\mathrm{TC}(p), \mathrm{DTC}(p)\}}. \tag{14}$$

The remaining tasks are to realize every coordinate obstruction with a genuine target law and to find the smallest one without enumerating schedules. Section 4 resolves both tasks and shows how the resulting coefficient composes across sequential blocks. Randomized cardinalities then change both the available decisions and, when their labels are hidden, the risk being optimized.

4 Exact deterministic robust schedules

For a fixed schedule, define its largest normalized load

$$M_{\mathcal{N}} = \max_{1 \leq i < n} \frac{d_{\mathcal{N}}(i)}{w_i}. \tag{15}$$

The jump identities immediately give

$$E_{\mathcal{N}}(p) \leq M_{\mathcal{N}} \min\{\mathrm{TC}(p), \mathrm{DTC}(p)\}. \tag{16}$$

Greedy reachability. Fix a candidate risk level $\rho \geq 0$. From a current batch boundary, the following recurrence takes the farthest next boundary allowed by that level. Starting at $u_1 = 1$, iterate

$$u_{r+1} = \min\{n+1, u_r + 1 + \lfloor \rho \min\{u_r, n - u_r\} \rfloor\}. \tag{17}$$

Let $G_n(\rho)$ be the number of generated nodes at most n.

Lemma 4.1 (Greedy feasibility)**.** A K-batch schedule with $M_{\mathcal{N}} \leq \rho$ exists if and only if $G_n(\rho) \leq K$.

Theorem 4.2 (Exact deterministic robust schedule)**.** For every $n \geq 3$, $1 \leq K \leq n$, and prime power $q \geq n - 1$,

$$\boxed{R^{(q)}_{n,K} = R_{n,K} = \min_{\mathcal{N}} \max_{1 \leq i < n} \frac{d_{\mathcal{N}}(i)}{w_i} = \inf\{\rho : G_n(\rho) \leq K\}.} \tag{18}$$

The greedy nodes at $\rho = R_{n,K}$, refined if necessary to exactly K batches, attain the minimum. For each fixed schedule, a maximizing index k is attained by the uniform law on an $[n, k]$ projective Reed–Solomon code. The upper guarantee holds over every finite alphabet; $q \geq n-1$ is the condition that makes all converse rays simultaneously realizable.

The theorem gives both the best guarantee and an attaining schedule. At $(n, K) = (10, 5)$, for example, an optimal schedule has batch sizes 1–2–3–3–1, whereas equal batches have size two. Both make five calls, but the largest normalized load falls from 1 to $2/3$.

Why the upper bound is sharp. Because every δ_i is nonnegative, Eqs. (9) to (11) bound the normalized error by $M_{\mathcal{N}}$. Conversely, the $[n, k]$ code law has projection entropy $H(X_A) = \min\{|A|, k\} \log q$. Its profile has a single nonzero jump at k, and

$$\mathrm{TC} = (n-k)\log q, \qquad \mathrm{DTC} = k \log q, \qquad E_{\mathcal{N}} = d_{\mathcal{N}}(k) \log q. \tag{19}$$

Its normalized loss is $d_{\mathcal{N}}(k)/w_k$, so a genuine target witnesses every exposed coordinate load.

Why the greedy construction is optimal. For consecutive schedule nodes $u < v$, the normalized load within that interval is largest at u, so

$$M_{\mathcal{N}} = \max_{a : N_a < n} \frac{N_{a+1} - N_a - 1}{\min\{N_a, n - N_a\}}. \tag{20}$$

Feasibility at level ρ is therefore equivalent to

$$v \leq \min\{n+1, u + 1 + \lfloor \rho \min\{u, n-u\} \rfloor\}. \tag{21}$$

The farthest feasible endpoint is monotone in u, so its iterates dominate every feasible prefix. It reaches $n+1$ in at most K batches exactly when $G_n(\rho) \leq K$, producing the shape in Figure 1.

Corollary 4.3 (One schedule controls both dependence budgets)**.** An optimal schedule $\mathcal{N}_*$ satisfies, for every non-product target,

$$E_{\mathcal{N}_*}(p) \leq R_{n,K} \min\{\mathrm{TC}(p), \mathrm{DTC}(p)\}, \tag{22}$$

and the coefficient cannot be reduced in the deterministic schedule class under the alphabet condition of Theorem 4.2.

Thus the same schedule is optimal whether the strongest dependence appears early or late in the reveal process.

Theorem 4.4 (Fast exact recovery)**.** The value and an attaining deterministic schedule can be recovered in $O(K \log n)$ exact arithmetic operations and $O(K)$ words of working memory. For $\epsilon > 0$, an absolute-ϵ certificate requires $O(K \log(2 + (n-K)/\epsilon))$ operations.

Every objective is a reduced rational with denominator at most $\lfloor n/2 \rfloor$. Dyadic feasibility tests shrink a bracket below their minimum separation, after which the feasible greedy witness identifies the optimum. See Appendix I for the proof and certificates.

Hiding a fixed coordinate order. The hidden-order output risk $\widetilde{E}_{\mathcal{N}}$ has the same minimax value. On every attaining code law, the sampler-to-target likelihood ratio is constant over target support and independent of the order.

Adaptive position selection. Within the exact-oracle, irrevocable, labelled game, fixing cardinalities means that policies based on revealed values or one-coordinate confidences cannot reduce the minimax value: on the code witnesses, unrevealed conditionals stay uniform until the codeword is determined, so the loss depends only on the batch sizes. Uniform order attains the upper bound; an arbitrary nonuniform ranking may incur more risk. See Theorems E.3 and E.4.

Sequential blocks. For blocks generated in a fixed sequence and unmasked internally in parallel, do local batching errors accumulate with the number of blocks? Write $\mathrm{TC}(X \mid C)$ and $\mathrm{DTC}(X \mid C)$ for the conditional dependence measures.

Corollary 4.5 (Exact sequential-block composition)**.** Let $X = (X^{(1)}, \dots, X^{(B)})$ be an ordered partition with block sizes $n_1, \dots, n_B$. Generate block r after conditioning on $H_r = (C, X^{(<r)})$, using a deterministic schedule $\mathcal{N}_r$ and a uniform within-block coordinate order. Define

$$M_r = \max_{1 \le i < n_r} \frac{d_{\mathcal{N}_r}(i)}{\min\{i, n_r - i\}}, \qquad M_{\max} = \max_r M_r. \tag{23}$$

Then the exact-oracle labelled loss obeys

$$E^{\mathrm{lab}}_{\mathrm{hier}}(p) = \sum_r \mathbb{E}_{H_r} E_{\mathcal{N}_r}(p_{X^{(r)}|H_r}) \le M_{\max} \min\{\mathrm{TC}(X \mid C), \mathrm{DTC}(X \mid C)\}. \tag{24}$$

The hidden-order output KL satisfies the same upper bound. If the ambient alphabet contains a prime-power subset of size at least $\max_r(n_r - 1)$, the normalized supremum of either loss equals $M_{\max}$. Consequently, for fixed per-block budgets,

$$R^{\mathrm{hier}}_{\mathbf{n},\mathbf{K}} = \max_r R_{n_r,K_r}. \tag{25}$$

The outer KL chain rule gives the sum in Eq. (24). The local conditional TC and DTC budgets sum to at most their global counterparts, and a single-jump law in the worst block proves sharpness. Thus the hardest block, not the number of blocks, sets the coefficient. The full proof, total-call allocation, and learned extension are in Appendix F.

Across batch budgets. The same recurrence yields all deterministic regimes. With few batches the value grows as $n^{1/K}$; for $\log n \ll K \ll n$, it is $\big(2 + o(1)\big) \log(n/K)/K$; at fixed call density it has an explicit floor-sensitive limit. Full statements and uniform inversions are in Theorem D.1.

5 Randomized schedules and learned predictors

Randomizing the batch sizes averages their load vectors, which can lower the largest exposed load. It also creates two distinct questions. The sampled schedule label may be retained while measuring KL, or it may be hidden by mixing the sampler outputs first. We analyze these risks separately.

5.1 Labelled risk: an exact finite game and sharp regimes

Let λ be a distribution over K-batch schedules, sampled before the target, and write $\bar{d}_\lambda(i) = \mathbb{E}_{\mathcal{N}\sim\lambda} d_{\mathcal{N}}(i)$. Under the same alphabet condition as Theorem 4.2, minimax duality and the single-jump code witnesses give, with Δ_{n-1} the probability simplex over the $n-1$ information indices,

$$R^{\mathrm{lab}}_{n,K} = \min_\lambda \max_{1 \le i < n} \frac{\bar{d}_\lambda(i)}{w_i} = \max_{\gamma \in \Delta_{n-1}} \min_{\mathcal{N}} \sum_{i=1}^{n-1} \gamma_i \frac{d_{\mathcal{N}}(i)}{w_i}. \tag{26}$$

The scheduler spreads load across schedules, while the adversary selects a dependence scale. The theorem quantifies the gain in three principal regimes.

Theorem 5.1 (Sharp gains from randomized batch sizes)**.** Let $W_n = \lfloor n^2/4 \rfloor$.

1. If $\log n \ll K \ll n$, then

$$R^{\mathrm{lab}}_{n,K} = (1 + o(1)) \frac{\log(n/K)}{K}, \qquad R^{\mathrm{det}}_{n,K} = (2 + o(1)) \frac{\log(n/K)}{K}. \tag{27}$$

2. If $K = n - m$, then

$$R^{\mathrm{lab}}_{n,K} = \frac{m}{W_n} \quad \Longleftrightarrow \quad m \le \left\lfloor \frac{n+1}{4} \right\rfloor. \tag{28}$$

   Consequently, if $K/n \to \beta \ge 3/4$, then $nR^{\mathrm{lab}}_{n,K} \to 4(1-\beta)$.

3. For $K = 2$, an explicit finite formula in Appendix G implies $R^{\mathrm{lab}}_{n,K} \sim \sqrt{n/2}$, compared with $R^{\mathrm{det}}_{n,K} \sim \sqrt{n}$.

The proof mechanisms differ. In the mesoscopic regime, a harmonic-scale adversary meets a phase-randomized logarithmic grid, halving the leading constant while retaining the $\log(n/K)$ universality price. Near the sequential limit, nonadjacent omitted boundaries produce the $3/4$ threshold; the two-batch formula uses a separate finite construction. See Appendix G. The same labelled minimax value holds for the transcript-adaptive policies in Theorem E.5.

5.2 Hidden schedule labels and output risk

The operational sampler does not reveal its random schedule. Its output risk is

$$R^{\mathrm{out}}_{n,K} = \min_{\lambda} \sup_{p \ne \mathrm{product}} \frac{D_{\mathrm{KL}}(p \| \mathbb{E}_{\mathcal{N},\pi} q_{\pi,\mathcal{N}})}{\min\{\mathrm{TC}(p), \mathrm{DTC}(p)\}}. \tag{29}$$

Jensen gives $R^{\mathrm{out}}_{n,K} \le R^{\mathrm{lab}}_{n,K}$, but average loads no longer determine the objective because output probabilities are mixed before KL. The smallest nontrivial case already exhibits the difference.

For $(n, K) = (3, 2)$, let A have widths $(1, 2)$ and B widths $(2, 1)$. After averaging the coordinate order, denote their output laws by $Q_A(p)$ and $Q_B(p)$, and put $q_t = tQ_A + (1-t)Q_B$. Define

$$V_{\mathcal{X}}(t) = \sup_{p \text{ non-product on } \mathcal{X}^3} \frac{D_{\mathrm{KL}}(p \| q_t)}{\min\{\mathrm{TC}(p), \mathrm{DTC}(p)\}}. \tag{30}$$

This is the worst output risk at a fixed mixture weight t.

Theorem 5.2 (Output risk of two-schedule mixtures)**.** For every finite alphabet $|\mathcal{X}| \ge 2$ and every $t \in [1/2, 1]$,

$$\boxed{V_{\mathcal{X}}(t) = t.} \tag{31}$$

For the binary global optimum, let ρ be the unique solution of $2^{1-\rho} = 1 + \rho$. Then

$$0.4569995591\ldots = \rho \le R^{\mathrm{out},(2)}_{3,2} \le \frac{1}{2}. \tag{32}$$

For $t \in [1/2, 1]$, the worst output risk is exact; optimizing the binary mixture is a separate nonlinear problem confined to Eq. (32). Jensen and three-variable TC/DTC identities give the fixed-law upper bound. A sparse boundary target preserves every proper marginal, so the $(1, 2)$ operator sees the corresponding product law while the $(2, 1)$ operator reproduces the target, giving the matching lower bound. Balancing this target with a common-bit boundary gives ρ; the symmetric mixture gives $1/2$. See Appendix H.

The role of the schedule label is now explicit. Retaining it turns randomization into a finite linear game with sharp gains, whereas hiding it mixes output probabilities before KL and produces the nonlinear value in Eq. (32).

5.3 From oracle conditionals to learned predictors

The results above use exact one-coordinate conditionals. Let $\widehat{p}_i(\cdot \mid X_S)$ be a learned predictor. On every target-positive context, assume it gives positive probability to every symbol having positive target conditional probability. For a uniform size-s set S and a uniform $i \notin S$, define

$$\varepsilon_s = \mathbb{E}_{S,i,X_S} D_{\mathrm{KL}}(p_i(\cdot \mid X_S) \,\|\, \widehat{p}_i(\cdot \mid X_S)). \tag{33}$$

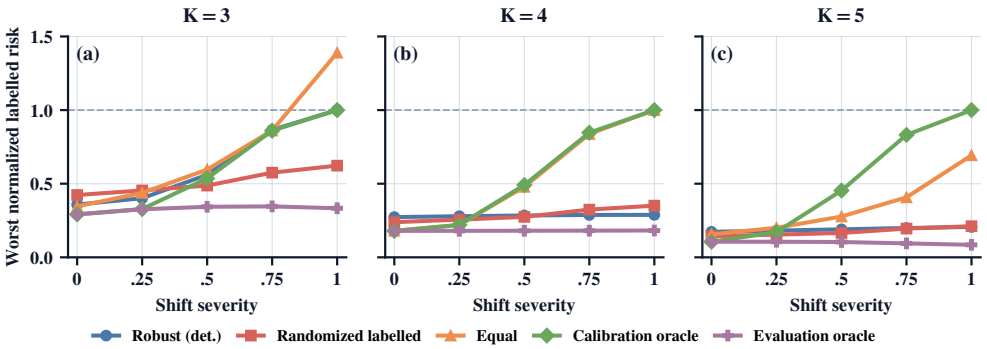

Figure 2: **Protection when dependence changes.** The calibration oracle uses the initial profile, and the evaluation oracle uses the shifted profile. Robust schedules use neither. Curves show the maximum normalized labelled risk across four target families. At large shifts, robust schedules reduce risk relative to the calibration oracle; at $K = 3$, this requires labelled randomization.

Under a uniform order, with batch sizes $b_a = N_{a+1} - N_a$, the decomposition into sampling and predictor error (Li & Cai, 2025; Chen et al., 2026; Lavenant & Zanella, 2025) takes the exact mask-level form

$$\widehat{E}^{\mathrm{lab}}_{\mathcal{N}}(p) = E_{\mathcal{N}}(p) + \sum_{a=1}^{K} b_a \varepsilon_{N_a - 1}. \tag{34}$$

The schedule controls the first term; predictor quality at visited mask levels controls the second. Both terms are exactly measurable for the enumerable finite laws used below. In language models, we hold the position-selection rule fixed and change batch sizes to measure their effect on generation accuracy.

6 Experiments

We test two claims. First, on finite laws whose risks can be computed exactly, the robust schedule of Section 4 keeps its guarantee when dependence shifts and when conditionals are learned, and target information can refine the choice. Second, in language models with the position rule held fixed, its batch sizes alone improve accuracy. Full protocols are in Appendix J.

Under shift, robust risk stays within its guarantee. To test what tuning to one dependence profile costs when dependence changes, we shift length-eight targets from a calibration law toward four endpoint laws. A calibration oracle fits the initial law, an evaluation oracle sees the shifted target, and the robust schedule uses neither; Figure 2 reports the largest normalized labelled risk. At maximum shift, calibration risk reaches one, while robust risks are 0.288 and 0.206 for $K = 4, 5$, below their guarantees $3/4$ and $1/2$. At $K = 3$, deterministic robust risk reaches its guarantee $R_{8,3} = 1$; labelled randomization lowers it to 0.623. Before the shift, the robust schedule exceeds the evaluation oracle by 0.067 to 0.094.

The advantage persists with learned conditionals. Practical decoders use learned conditionals, whose errors add a second term to the risk in Eq. (34). We train masked neural predictors on length-eight binary laws and compare robust and equal batch sizes, using enumeration to compute the exact KL of the generated distribution. Across 36 combinations of targets and training seeds, robust batching lowers mean output KL from 0.121 to 0.069 and worst normalized risk from 1.065 to 0.199 relative to equal batching.

With target information, oracle risk selects good schedules. When the target family is known, a schedule can be chosen by its oracle output risk; we test whether this choice holds up with learned conditionals. On a separate, prespecified set of 12 target laws and three new training seeds, we rank all 35 schedules with four nonempty batches. Oracle and learned output risks have Spearman correlation 0.917. Under worst normalized learned output KL, the oracle-selected schedule ranks

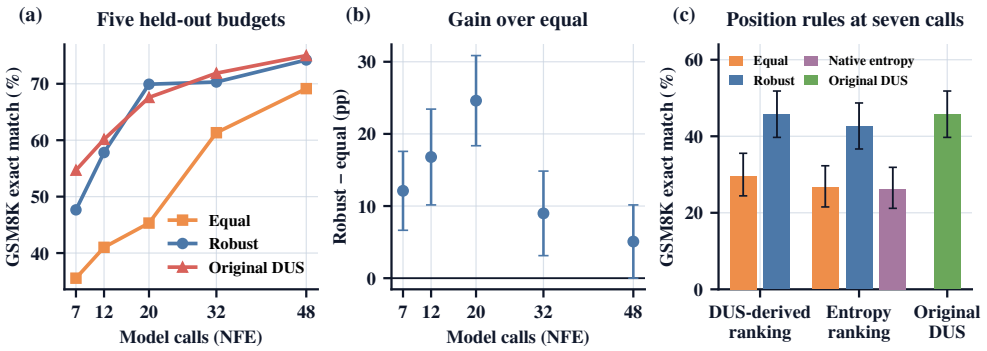

Figure 3: **Batch-size effects across budgets and position policies.** (a) Equal and robust batching share a DUS-derived position ranking within each configuration on 256 held-out GSM8K prompts. Original DUS is the complete system reference. (b) Paired accuracy gains over equal batching with 95% intervals. (c) At about seven calls on a separate 256-prompt split, equal and robust batching share either a frozen DUS-derived ranking or an entropy ranking; native entropy-adaptive batch sizes and original DUS are references.

third; on this family, equal batching ranks first and the robust schedule 19th. The minimax criterion protects against arbitrary targets, whereas oracle output risk selects schedules for a specified family.

Robust batch sizes improve LLaDA across budgets. We next ask whether batch sizes alone change accuracy in a language model. We run LLaDA-8B-Instruct (Nie et al., 2025) on 256 held-out GSM8K prompts (Cobbe et al., 2021), generating 128 tokens with $7, 12, 20, 32, 48$ model calls (NFE). Equal and robust batching share the position ranking of the DUS decoder (Luxembourg et al., 2026) and differ only in batch boundaries; block sizes follow DUS's base-2 setting, with $\log_2 b$ calls per block of b tokens. Robust batching improves exact match by 5.08 to 24.61 points, peaking at 20 calls and shrinking at the two highest budgets (Figure 3a,b). Changing only batch boundaries recovers 63–88% of DUS's gain over equal batching at four budgets and exceeds DUS at 20 calls.

The gain does not rely on the DUS ranking. To check that the gain is not specific to one ranking, we change the position rule. Under uniformly random positions, the setting of our theory, robust batching at 32 calls raises exact match from 46.88% to 61.72% on LLaDA and from 26.56% to 38.28% on Dream-7B-Instruct (Ye et al., 2025), with the same runtime and peak memory. With a dynamic entropy-based position rule at seven calls, it improves accuracy by 16.02 points over equal batching and by 16.41 points over native entropy-adaptive batch sizes at comparable mean call counts (Figure 3c).

Where the boundaries fall matters. Finally, we test whether any unequal schedule would do. At 32 calls under uniform positions, robust batching outperforms all tested tapered, reversed, and shuffled schedules. Reversed and shuffled schedules use the same batch sizes in a different order, yet the best control, the reversed one, still trails by 9.64 points.

7 Conclusion

Parallel unmasking still requires a batch-size decision when reliable dependency information is unavailable. We characterize the exact finite limit for deterministic batch sizes, construct schedules that attain it, and give matching target laws. Across sequential blocks, the hardest block sets the coefficient. Randomization improves labelled risk, while hiding the schedule makes output risk nonlinear. We characterize this risk for a range of mixture weights and bound the optimum for binary targets. The learned-predictor decomposition and controlled experiments show that batch size remains consequential after the position ranking is fixed. The resulting boundary gives target-aware and adaptive decoders a precise reference for separating gains from dependency information, position selection, and batch allocation.

AI USE STATEMENT

In this work, we used generative AI tools for language polishing and improving readability. The authors reviewed the AI-assisted revisions and take responsibility for the final content of this work.

REFERENCES

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/958c530554f78bcd8e97125b70e6973d-Abstract.html.

Iskander Azangulov, Teodora Pandeva, Niranjani Prasad, Javier Zazo, and Sushrut Karmalkar. Parallel sampling from masked diffusion models via conditional independence testing. *arXiv preprint arXiv:2510.21961*, 2025. URL https://arxiv.org/abs/2510.21961.

Parikshit Bansal and Sujay Sanghavi. Enabling approximate joint sampling in diffusion LMs. *arXiv preprint arXiv:2509.22738*, 2025. URL https://arxiv.org/abs/2509.22738.

Vansh Bansal, Cho Cholyeon, Syamantak Kumar, Sujay Sanghavi, and Purnamrita Sarkar. Understanding parallel samplers in masked diffusion via random walks on graphs. *arXiv preprint arXiv:2606.22976*, 2026. URL https://arxiv.org/abs/2606.22976.

Heli Ben-Hamu, Itai Gat, Daniel Severo, Niklas Nolte, and Brian Karrer. Accelerated sampling from masked diffusion models via entropy bounded unmasking. *arXiv preprint arXiv:2505.24857*, 2025. URL https://arxiv.org/abs/2505.24857.

Shirui Chen, Jiantao Jiao, Lillian J. Ratliff, and Banghua Zhu. dUltra: Ultra-fast diffusion language models via reinforcement learning. *arXiv preprint arXiv:2512.21446*, 2025a. URL https://arxiv.org/abs/2512.21446.

Sitan Chen, Kevin Cong, and Jerry Li. Optimal inference schedules for masked diffusion models. In *Proceedings of Thirty-Ninth Conference on Learning Theory*, volume 336 of *Proceedings of Machine Learning Research*, pp. 1279–1311. PMLR, 2026.

Zigeng Chen, Gongfan Fang, Xinyin Ma, Ruonan Yu, and Xinchao Wang. dParallel: Learnable parallel decoding for dLLMs. *arXiv preprint arXiv:2509.26488*, 2025b. URL https://arxiv.org/abs/2509.26488.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Rei Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. URL https://arxiv.org/abs/2110.14168.

Daniil Dmitriev, Zhihan Huang, and Yuting Wei. Provably adaptive sampling with uniform and remasking discrete diffusion models. *arXiv preprint arXiv:2608.23554*, 2026. URL https://arxiv.org/abs/2608.23554.

Metod Jazbec, Theo X. Olausson, Louis Béthune, Pierre Ablin, Michael Kirchhof, João Monteiro, Victor Turrisi, Jason Ramapuram, and Marco Cuturi. Learning unmasking policies for diffusion language models. *arXiv preprint arXiv:2512.09106*, 2025. URL https://arxiv.org/abs/2512.09106.

Haozhe Jiang, Nika Haghtalab, and Lijie Chen. Diffusion language models are provably optimal parallel samplers. *arXiv preprint arXiv:2512.25014*, 2025. URL https://arxiv.org/abs/2512.25014.

Bumjun Kim, Dongjae Jeon, Moongyu Jeon, and Albert No. Dependency-aware parallel decoding via attention for diffusion LLMs. *arXiv preprint arXiv:2603.12996*, 2026. URL https://arxiv.org/abs/2603.12996.

Hugo Lavenant and Giacomo Zanella. Error bounds and optimal schedules for masked diffusions with factorized approximations. *arXiv preprint arXiv:2510.25544*, 2025. URL https://arxiv.org/abs/2510.25544.

Gen Li and Changxiao Cai. Breaking AR's sampling bottleneck: Provable acceleration via diffusion language models. *arXiv preprint arXiv:2505.21400*, 2025. URL https://arxiv.org/abs/2505.21400.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 32819–32848. PMLR, 2024. URL https://proceedings.mlr.press/v235/lou24a.html.

Lizhuo Luo, Shenggui Li, Yonggang Wen, and Tianwei Zhang. DSB: Dynamic sliding block scheduling for diffusion LLMs. *arXiv preprint arXiv:2602.05992*, 2026a. URL https://arxiv.org/abs/2602.05992.

Lizhuo Luo, Zhuoran Shi, Jiajun Luo, Zhi Wang, Shen Ren, Wenya Wang, and Tianwei Zhang. DAWN: Dependency-aware fast inference for diffusion LLMs. *arXiv preprint arXiv:2602.06953*, 2026b. URL https://arxiv.org/abs/2602.06953.

Omer Luxembourg, Haim Permuter, and Eliya Nachmani. Plan for speed: Dilated scheduling for masked diffusion language models. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2506.19037.

Amr Mohamed, Yang Zhang, Michalis Vazirgiannis, and Guokan Shang. Fast-decoding diffusion language models via progress-aware confidence schedules. *arXiv preprint arXiv:2512.02892*, 2025. URL https://arxiv.org/abs/2512.02892.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. *arXiv preprint arXiv:2502.09992*, 2025. URL https://arxiv.org/abs/2502.09992.

Irving S. Reed and Gustave Solomon. Polynomial codes over certain finite fields. *Journal of the Society for Industrial and Applied Mathematics*, 8(2):300–304, 1960. doi: 10.1137/0108018.

Liran Ringel, Ameen Ali, and Yaniv Romano. Dependency-guided parallel decoding in discrete diffusion language models. *arXiv preprint arXiv:2604.02560*, 2026. URL https://arxiv.org/abs/2604.02560.

Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T. Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. *arXiv preprint arXiv:2406.07524*, 2024. URL https://arxiv.org/abs/2406.07524.

Yangyi Shen, Tianjian Feng, Jiaqi Han, Wen Wang, Tianlang Chen, Chunhua Shen, Jure Leskovec, and Stefano Ermon. Improving diffusion language model decoding through joint search in generation order and token space. *arXiv preprint arXiv:2601.20339*, 2026. URL https://arxiv.org/abs/2601.20339.

Shengyin Sun, Yiming Li, Renxi Liu, Xinqi Li, Hui-Ling Zhen, Weizhe Lin, Chen Chen, Xianzhi Yu, Mingxuan Yuan, and Chen Ma. PSD: Pushing the pareto frontier of diffusion LLMs via parallel speculative decoding. *arXiv preprint arXiv:2605.15609*, 2026. URL https://arxiv.org/abs/2605.15609.

Martin J. Wainwright. The data geometry of masking diffusion: Certified-optimal schedules via unmasking growth complexity. *arXiv preprint arXiv:2608.13520*, 2026. URL https://arxiv.org/abs/2608.13520.

Xinyun Wang, Min Zhang, Sen Cui, Zhikang Chen, Bo Jiang, Kun Kuang, and Mingbao Lin. Reversible diffusion decoding for diffusion language models. *arXiv preprint arXiv:2602.00150*, 2026. URL https://arxiv.org/abs/2602.00150.

Xu Wang, Chenkai Xu, Yijie Jin, Jiachun Jin, Hao Zhang, and Zhijie Deng. Diffusion LLMs can do faster-than-AR inference via discrete diffusion forcing. *arXiv preprint arXiv:2508.09192*, 2025. URL https://arxiv.org/abs/2508.09192.

Yanzheng Xiang, Lan Wei, Yizhen Yao, Qinglin Zhu, Hanqi Yan, Chen Jin, Philip Alexander Teare, Dandan Zhang, Lin Gui, Amrutha Saseendran, and Yulan He. Stop the flip-flop: Context-preserving verification for fast revocable diffusion decoding. *arXiv preprint arXiv:2602.06161*, 2026. URL https://arxiv.org/abs/2602.06161.

Jiawei Xu, Minghui Liu, Aakriti Agrawal, Yifan Chen, and Furong Huang. Scheduling thoughts: Learning the order of thought in diffusion language models. *arXiv preprint arXiv:2606.23567*, 2026. URL https://arxiv.org/abs/2606.23567.

Jiacheng Ye et al. Dream 7b: Diffusion large language models. *arXiv preprint arXiv:2508.15487*, 2025. URL https://arxiv.org/abs/2508.15487.

Yu Zhang, Xinchen Li, Jialei Zhou, Hongnan Ma, Zhongwei Wan, Yiwei Shi, Duoqian Miao, Qi Zhang, and Longbing Cao. Swordsman: Entropy-driven adaptive block partition for efficient diffusion language models. *arXiv preprint arXiv:2602.04399*, 2026. URL https://arxiv.org/abs/2602.04399.

Yunxiao Zhao and Changxiao Cai. Adaptation to intrinsic dependence in diffusion language models. *arXiv preprint arXiv:2602.20126*, 2026. URL https://arxiv.org/abs/2602.20126.

A RISK IDENTITIES AND THE INFORMATION PROFILE

This appendix derives the exact loss formula used in the paper and records the entropy identities behind the information profile.

A.1 BLOCK FACTORIZATION KL

Fix a permutation π and a schedule $\mathcal{N}$. The expression $q_{\pi,\mathcal{N}}$ in Eq. (2) is a probability mass function: conditionally on all earlier batches, the product over a current batch is a product probability law, and summing batch by batch gives total mass one. All conditional probabilities below are evaluated only on contexts having positive p-probability, which suffices for expectations under p.

Factor p by the chain rule along the same permutation:

$$p(x) = \prod_{j=1}^{n} p(x_{\pi_j} \mid x_{\pi_1}, \ldots, x_{\pi_{j-1}}). \tag{35}$$

For $N_a \le j < N_{a+1}$, the contribution of position j to the forward KL from p to the block law is therefore

$$\begin{aligned} &\mathbb{E}_p\left[\log \frac{p(X_{\pi_j} \mid X_{\pi_1}, \ldots, X_{\pi_{j-1}})}{p(X_{\pi_j} \mid X_{\pi_1}, \ldots, X_{\pi_{N_a-1}})}\right] \\ &\quad = H(X_{\pi_j} \mid X_{\pi_1}, \ldots, X_{\pi_{N_a-1}}) - H(X_{\pi_j} \mid X_{\pi_1}, \ldots, X_{\pi_{j-1}}). \end{aligned} \tag{36}$$

Let

$$\overline{H} := \frac{1}{n}\sum_{r=1}^{n} H(X_r).$$

If π is uniform, then at position j its prefix is a uniform set S of size $j-1$, and π_j is uniform outside S. Consequently,

$$\mathbb{E}_\pi H(X_{\pi_j} \mid X_{\pi_1}, \ldots, X_{\pi_{j-1}}) = \overline{H} - Z_j. \tag{37}$$

The same argument with a prefix of size $N_a - 1$ gives $\overline{H} - Z_{N_a}$ for the first entropy in Eq. (36). Write $Z_j^{\mathcal{N}} = Z_{N_a}$ for $N_a \le j < N_{a+1}$. Summing over positions and batches proves

Proposition A.1 (Expected labelled-order KL)**.** For every target p and deterministic schedule $\mathcal{N}$,

$$\begin{aligned} E_{\mathcal{N}}(p) &= \sum_{a=1}^{K} \sum_{j=N_a}^{N_{a+1}-1} (Z_j - Z_{N_a}) \\ &= \sum_{j=1}^{n} (Z_j - Z_j^{\mathcal{N}}). \end{aligned} \tag{38}$$

An artificial ordering of coordinates inside a batch does not change $q_{\pi,\mathcal{N}}$: every factor in that batch conditions on the same earlier coordinates. The permutation is only a convenient representation of the uniformly random labelled partition. This proves that Eq. (38) evaluates the risk in Eq. (3), rather than a mixture distribution over permutations.

A.2 MONOTONICITY AND DEPENDENCE IDENTITIES

For a fixed coordinate r, draw a uniform random ordering of the other $n-1$ coordinates and let S_m contain its first m entries. Then S_m is a uniform m-subset and $S_m \subset S_{m+1}$. The chain rule for mutual information gives

$$I(X_r; X_{S_{m+1}}) = I(X_r; X_{S_m}) + I(X_r; X_{S_{m+1}\setminus S_m} \mid X_{S_m}) \geq I(X_r; X_{S_m}). \tag{39}$$

Averaging first over the ordering and then over r proves $Z_{j+1} \geq Z_j$, and hence Eq. (7).

Average the chain-rule entropy of X over a uniform permutation and apply Eq. (37):

$$H(X_1, \ldots, X_n) = \sum_{j=1}^{n} (\overline{H} - Z_j) = \sum_{r=1}^{n} H(X_r) - \sum_{j=1}^{n} Z_j. \tag{40}$$

Subtracting from the sum of marginal entropies proves the first identity in Eq. (12). At the last profile location,

$$nZ_n = \sum_{r=1}^{n} I(X_r; X_{[n]\setminus\{r\}}) = \sum_{r=1}^{n} \big[H(X_r) - H(X_r \mid X_{[n]\setminus\{r\}})\big]. \tag{41}$$

Subtracting $\mathrm{TC}(p)$ from Eq. (41) gives exactly Eq. (5), proving the second identity in Eq. (12).

A.3 PROOF OF THE JUMP REPRESENTATION

Since $Z_1 = 0$, telescoping gives

$$Z_j = \sum_{i=1}^{j-1} \delta_i. \tag{42}$$

Summing Eq. (42) over j and reversing the order gives

$$\sum_{j=1}^{n} Z_j = \sum_{i=1}^{n-1} (n-i)\delta_i. \tag{43}$$

Similarly, $Z_n = \sum_{i=1}^{n-1} \delta_i$, so

$$nZ_n - \sum_{j=1}^{n} Z_j = \sum_{i=1}^{n-1} i\delta_i. \tag{44}$$

Finally, within a schedule interval $N_a \leq j < N_{a+1}$,

$$Z_j - Z_{N_a} = \sum_{i=N_a}^{j-1} \delta_i. \tag{45}$$

A fixed δ_i with $N_a \leq i < N_{a+1}$ occurs in Eq. (45) for $j = i+1, \ldots, N_{a+1}-1$, exactly $N_{a+1}-i-1 = d_{\mathcal{N}}(i)$ times. Substitution in Eq. (38) proves Eq. (9); Eqs. (43) and (44) prove the other two parts of Lemma 3.1.

As a useful nondegeneracy consequence, $\mathrm{TC}(p)$ is the KL from p to the product of its marginals and vanishes exactly when p is a product law. If p is not a product, Eq. (10) forces some $\delta_i > 0$. Every coefficient i in Eq. (11) is positive, so $\mathrm{DTC}(p) > 0$.

B GREEDY COVERING AND EXACT COMPUTATION

We first solve the optimization over nonnegative information jumps and then solve the remaining integer schedule problem.

B.1 THE NONNEGATIVE JUMP CONE

For a fixed schedule, recall $M_{\mathcal{N}}$ from Eq. (15). For every nonzero vector $\delta \in \mathbb{R}_+^{n-1}$,

$$\begin{aligned}\sum_i d_{\mathcal{N}}(i)\delta_i &\le M_{\mathcal{N}} \sum_i w_i \delta_i \\ &\le M_{\mathcal{N}} \min\left\{ \sum_i i\delta_i, \sum_i (n-i)\delta_i \right\}. \end{aligned} \tag{46}$$

Conversely, if δ is supported only at an index k, its normalized ratio is $d_{\mathcal{N}}(k)/w_k$. Choosing a maximizing k proves the cone identity

$$\sup_{\delta \in \mathbb{R}_+^{n-1} \setminus \{0\}} \frac{\sum_i d_{\mathcal{N}}(i)\delta_i}{\min\{\sum_i i\delta_i, \sum_i (n-i)\delta_i\}} = M_{\mathcal{N}}. \tag{47}$$

Appendix C will verify that each maximizing single-coordinate vector in Eq. (47) is realized by a target in Eq. (14).

B.2 THE WORST INDEX OF AN INTERVAL

Let $u < v$ be consecutive schedule nodes, put $D = v - u - 1$, and consider a valid information index $i = u + t < n$, where $0 \le t \le D$. Its numerator is $d_{\mathcal{N}}(i) = D - t$. We prove

$$\frac{D-t}{w_{u+t}} \le \frac{D}{w_u}. \tag{48}$$

There are three cases. If $u + t \le n/2$, the numerator weakly decreases with t while $w_{u+t} = u + t$ weakly increases, proving Eq. (48).

Suppose next that $u \le n/2 \le u + t$, and write $A = n - u$. Then $w_u = u$, $w_{u+t} = A - t$, and the cross-multiplied inequality is

$$u(D - t) \le D(A - t). \tag{49}$$

The right side minus the left side equals

$$D(A - u) + t(u - D). \tag{50}$$

If $D \le u$, both terms in Eq. (50) are nonnegative. If $D > u$, the expression is minimized over $0 \le t \le D$ at $t = D$, where it equals $D(A - D) \ge 0$. The last inequality follows from $v \le n + 1$, which implies $D \le n - u = A$.

Finally, suppose $u \ge n/2$. Then $w_u = A = n - u$, $w_{u+t} = A - t > 0$, and $D \le A$. Cross multiplication shows

$$\frac{D-t}{A-t} \le \frac{D}{A} \quad \Longleftrightarrow \quad tD \le tA, \tag{51}$$

which again follows from $D \le A$. This completes all cases and proves Eq. (48). Taking the maximum over schedule intervals gives Eq. (20).

B.3 GREEDY FEASIBILITY AND ATTAINMENT

For an interval starting at $u < n$, the constraint in Eq. (20) is

$$\frac{v - u - 1}{w_u} \le \rho. \tag{52}$$

Because the numerator is an integer, Eq. (52) is equivalent to

$$v \le F_\rho(u) := \min\{n + 1, u + 1 + \lfloor \rho w_u \rfloor\}, \tag{53}$$

which is Eq. (21).

We next verify that F_ρ is nondecreasing on integer nodes $1, \dots, n$. On the increasing half of w_u, both $u + 1$ and $\lfloor \rho w_u \rfloor$ are nondecreasing. If $0 \le \rho < 1$, moving one step along the decreasing half reduces w_u by at most one and hence reduces its floored multiple by at most one; the simultaneous increment of $u + 1$ offsets this reduction. If $\rho \ge 1$, every $u \ge \lceil n/2 \rceil$ satisfies

$$u + 1 + \lfloor \rho(n - u) \rfloor \ge n + 1,$$

so $F_\rho(u) = n + 1$ throughout the decreasing half. The midpoint transition is covered by the same observations.

Let $u_1, u_2, \dots$ be the greedy sequence $u_{r+1} = F_\rho(u_r)$. Suppose $N_1, \dots, N_{K+1}$ is any schedule satisfying $M_\mathcal{N} \le \rho$. We prove by induction that $N_r \le u_r$ until the terminal node. Equality holds at $r = 1$. If $N_r \le u_r$, feasibility and monotonicity give

$$N_{r+1} \le F_\rho(N_r) \le F_\rho(u_r) = u_{r+1}. \tag{54}$$

At $r = K + 1$, the left side is $n + 1$, the largest possible node. The greedy sequence must therefore reach $n + 1$ in at most K transitions, which is $G_n(\rho) \le K$.

Conversely, suppose $G_n(\rho) = m \le K$. The m greedy nodes in $[n]$, followed by $n + 1$, satisfy Eq. (53) and hence form a feasible m-batch schedule. There are $n - m \ge K - m$ unused integer nodes in $[n]$, so insert any $K - m$ of them in increasing order. Splitting an interval moves a next boundary weakly left for indices before the inserted node and leaves later next boundaries unchanged. Thus every $d_\mathcal{N}(i)$, and hence $M_\mathcal{N}$, can only decrease. This proves Lemma 4.1 in both directions.

There are finitely many schedules in $\mathfrak{N}_{n,K}$, so

$$\rho_* := \min_{\mathcal{N} \in \mathfrak{N}_{n,K}} M_\mathcal{N} \tag{55}$$

is attained. The feasibility equivalence shows that $G_n(\rho_*) \le K$, while $G_n(\rho) > K$ for every $\rho < \rho_*$. Therefore

$$\rho_* = \inf\{\rho \ge 0 : G_n(\rho) \le K\}. \tag{56}$$

Using the greedy nodes at ρ_* and refining to exactly K nodes gives the schedule asserted in Theorem 4.2.

B.4 EXACT ALGORITHM AND EXHAUSTIVE FINITE VERIFICATION

Every deterministic optimum is one of the rational numbers

$$\mathcal{C}_n := \left\{ \frac{a}{w_u} : 1 \le u < n, \quad 0 \le a \le n - u \right\}. \tag{57}$$

Indeed, the objective of a fixed schedule is the maximum of its interval left-endpoint ratios, and each such ratio belongs to $\mathcal{C}_n$; minimizing over finitely many schedules preserves membership. The multiset in Eq. (57) has

$$\sum_{u=1}^{n-1}(n-u+1) = \frac{n(n+1)}{2} - 1$$

entries before duplicate removal.

The following complete pseudocode computes both the exact reduced fraction and a minimizing schedule.

1. Generate $\mathcal{C}_n$, reduce each fraction, remove duplicates, and sort it by cross multiplication.
2. Binary-search the sorted list. For a candidate $\rho = a/b$, start at $u = 1$, repeatedly replace
$$u \leftarrow \min\{n+1, u+1+\lfloor a\, w_u/b \rfloor\},$$
and count the nodes in $[n]$. Declare the candidate feasible exactly when the count is at most K.
3. Return the first feasible candidate. Run the recurrence once more at that value, and repeatedly insert the smallest unused node in $[n]$ until exactly K batch starts remain.

Monotonicity of G_n and Lemma 4.1 prove correctness. The candidate generation and sorting use $O(n^2 \log n)$ rational comparisons; binary search adds $O(n \log n)$ integer updates. Every input fraction has numerator and denominator at most n, and every floor is evaluated as an exact integer quotient, so all intermediate integers have $O(\log n)$ bits apart from a single $O(\log n)$-bit product. This gives polynomial bit complexity and makes the floor convention reproducible without floating point. The dyadic search in Appendix I recovers the same optimum without enumerating $\mathcal{C}_n$, using $O(K \log n)$ exact arithmetic operations.

As an independent indexing check, one can enumerate every choice of the $K-1$ internal starts among $\{2, \ldots, n\}$, evaluate Eq. (20), and compare its minimum to the recurrence threshold. The complete results through $n = 8$ are:

n	$K = 1, \ldots, n$	exact values $R_{n,K}$
3	$1, 2, 3$	$2,\ 1,\ 0$
4	$1, 2, 3, 4$	$3,\ 1,\ 1/2,\ 0$
5	$1, 2, 3, 4, 5$	$4,\ 1,\ 1,\ 1/2,\ 0$
6	$1, 2, 3, 4, 5, 6$	$5,\ 1,\ 1,\ 1/2,\ 1/3,\ 0$
7	$1, 2, 3, 4, 5, 6, 7$	$6,\ 4/3,\ 1,\ 2/3,\ 1/2,\ 1/3,\ 0$
8	$1, 2, 3, 4, 5, 6, 7, 8$	$7,\ 5/3,\ 1,\ 3/4,\ 1/2,\ 1/3,\ 1/4,\ 0$

For each row and each K, the exhaustive search covers exactly $\binom{n-1}{K-1}$ schedules. At every entry, exhaustive enumeration and the greedy recurrence give the same value.

C Projective Reed–Solomon realization of the rays

The upper bound in Eq. (46) treats every nonnegative jump vector as possible. Code laws with single-jump profiles are used by Chen et al. (2026). Here the projectively extended Reed–Solomon construction (Reed & Solomon, 1960) realizes every ray for lengths up to $q+1$.

Fix $k \in \{1, \ldots, n-1\}$. Since $q \geq n-1$, choose distinct points $\alpha_1, \ldots, \alpha_n$ in the projective line $\mathbb{F}_q \cup \{\infty\}$, which has $q+1$ points. For $f(t) = \sum_{j=0}^{k-1} f_j t^j$, use the convention $f(\infty) := f_{k-1}$, and let

$$C_k := \{(f(\alpha_1), \ldots, f(\alpha_n)) : f \in \mathbb{F}_q[t],\ \deg f < k\}, \tag{58}$$

and let $p_k = \mathrm{Unif}(C_k)$.

Lemma C.1 (Entropy of code projections)**.** For every coordinate set $A \subseteq [n]$,

$$H_{p_k}(X_A) = \min\{|A|, k\} \log q. \tag{59}$$

Proof. The coefficient vector $(f_0, \dots, f_{k-1})$ is uniform on $\mathbb{F}_q^k$, and the length-n evaluation map is linear. It is injective because the argument below gives rank k for the full coordinate set; hence C_k has exactly q^k codewords and dimension k. The map taking a polynomial of degree below k to its evaluations on A is linear. If $\infty \notin A$, ordinary interpolation proves surjectivity for $|A| \le k$, and k finite roots prove injectivity for $|A| \ge k$. If $\infty \in A$ and $|A| \le k$, first prescribe f_{k-1}; a polynomial of degree below $k-1$ then interpolates the residual values at the other $|A| - 1 \le k - 1$ finite points. If $\infty \in A$, $|A| \ge k$, and all evaluations vanish, then $f_{k-1} = 0$, so f has degree at most $k-2$ and at least $k-1$ finite roots; hence $f = 0$. Thus the projection rank is $\min\{|A|, k\}$ in every case. The image is therefore a linear space of size $q^{\min\{|A|,k\}}$. A uniform input polynomial induces the uniform law on this image, proving Eq. (59). □

Thus every puncturing to m coordinates has dimension $\min\{m, k\}$; in coding terminology C_k is an $[n, k, n-k+1]$ MDS code. This includes both load-bearing endpoints: for $k = 1$, every nonempty projection has entropy $\log q$, while for $k = n-1$, every $(n-1)$-coordinate projection is a bijective image of $\mathbb{F}_q^{n-1}$ and the full vector has exactly one linear constraint. The construction uses all $q+1$ projective points when $n = q+1$, so the asserted length remains valid at the boundary $q = n-1$.

Lemma C.2 (Conditional fibers of the projective code)**.** Let $S, B \subseteq [n]$ be disjoint, put $|S| = s$, and condition on any x_S having positive p_k-probability. Then

$$H_{p_k}(X_B \mid X_S = x_S) = \big(\min\{s + |B|, k\} - \min\{s, k\}\big) \log q. \tag{60}$$

In particular, if $s < k$, every unrevealed one-coordinate conditional is uniform on $\mathbb{F}_q$, and the joint continuation on a batch of size b is uniform on an affine space of size $q^{\min\{b, k-s\}}$. If $s \ge k$, the entire codeword is determined.

Proof. The fiber of coefficient vectors consistent with x_S is a nonempty affine translate of the kernel of the projection to S. By Lemma C.1, the ranks of the projections to S and $S \cup B$ are respectively $\min\{s, k\}$ and $\min\{s + |B|, k\}$. The induced image of that affine fiber on B therefore has dimension equal to their rank difference. Uniformity of the coefficient vector makes the conditional law uniform on this image, proving Eq. (60) and all stated cases. □

Take $|S| = j - 1$ and $r \notin S$. By Lemma C.1,

$$\begin{aligned} I_{p_k}(X_r; X_S) &= H(X_r) + H(X_S) - H(X_{S \cup \{r\}}) \\ &= \big[1 + \min\{j-1, k\} - \min\{j, k\}\big] \log q \\ &= \begin{cases} 0, & j \le k, \\ \log q, & j \ge k+1. \end{cases} \end{aligned} \tag{61}$$

This value is independent of the particular admissible pair (S, r), so averaging gives

$$Z_j(p_k) = \begin{cases} 0, & j \le k, \\ \log q, & j \ge k+1. \end{cases} \tag{62}$$

Thus the only nonzero profile jump is

$$\delta_k = \log q. \tag{63}$$

Substitution into Lemma 3.1 proves the three values in Eq. (19). Because $1 \le k < n$, both $(n-k)\log q$ and $k \log q$ are positive. Hence p_k belongs to the target class in Eq. (14).

We can now complete the distributional minimax argument. For every schedule, Eq. (16) gives

$$\sup_p \frac{E_{\mathcal{N}}(p)}{\min\{\mathrm{TC}(p), \mathrm{DTC}(p)\}} \le M_{\mathcal{N}}, \tag{64}$$

where the supremum is over the admissible laws in Eq. (14). Choose k attaining the maximum in $M_{\mathcal{N}}$. For p_k, Eq. (19) gives

$$\frac{E_{\mathcal{N}}(p_k)}{\min\{\mathrm{TC}(p_k), \mathrm{DTC}(p_k)\}} = \frac{d_{\mathcal{N}}(k)}{\min\{k, n-k\}} = M_{\mathcal{N}}. \tag{65}$$

Therefore equality holds in Eq. (64) for each individual schedule. Minimizing over schedules and applying Eq. (56) proves Theorem 4.2. Applying the minimizing schedule to Eq. (16) proves Corollary 4.3.

D Deterministic phase diagram

All batch-count regimes. Define the strictly decreasing function

$$\Psi(r) := \begin{cases} \dfrac{1}{\log(1+r)} + \dfrac{1}{-\log(1-r)}, & 0 < r < 1, \\ \dfrac{1}{\log(1+r)}, & r \geq 1, \end{cases} \tag{66}$$

and the continuous strictly decreasing function on $c \in [2, \infty)$

$$\Phi(c) := 2 \int_0^{1/2} \frac{dx}{1 + \lfloor cx \rfloor}. \tag{67}$$

Here $\Phi(2) = 1$ and $\Phi(c) \downarrow 0$, so the inverse used below is well defined.

Theorem D.1 (Deterministic phase diagram)**.** Let $n \to \infty$ and $1 \leq K = K(n) \leq n$.

(i) If $K = o(\log n)$, then

$$R_{n,K} = (1 + o(1))n^{1/K}.$$

(ii) If $K/\log n \to \kappa \in (0, \infty)$, then

$$R_{n,K} \longrightarrow r_\kappa, \qquad \Psi(r_\kappa) = \kappa.$$

(iii) If $K/\log n \to \infty$ and $K/n \to 0$, then

$$R_{n,K} = (2 + o(1))\frac{\log(n/K)}{K}.$$

(iv) If $K/n \to \beta \in (0, 1)$, then

$$nR_{n,K} \longrightarrow c_\beta, \qquad \Phi(c_\beta) = \beta.$$

(v) If $K < n$ eventually and $K/n \to 1$, then

$$nR_{n,K} \longrightarrow 2.$$

For the remaining endpoint $K = n$, $R_{n,n} = 0$ exactly.

All $o(1)$ terms are uniform along sequences satisfying the conditions of their respective case.

For example, if $\log K/\log n \to \alpha \in [0, 1)$ while $K/\log n \to \infty$, part (iii) becomes

$$R_{n,K} = \big(2(1-\alpha) + o(1)\big)\frac{\log n}{K}. \tag{68}$$

Thus the constant 2 in the subpolynomial range is the $\alpha = 0$ member of a continuum of constants, rather than an isolated approximation. The proofs below derive every phase from the integer recurrence, retaining its floors and accounting for the endpoint tails and midpoint crossing.

D.1 UNIFORM HITTING-TIME LEMMAS

The following estimates cover every scale needed in Theorem D.1.

Lemma D.2 (Greedy hitting times)**.** Let G_n be defined by Eq. (17).

(a) For every fixed $\rho > 0$,

$$\frac{G_n(\rho)}{\log n} \longrightarrow \Psi(\rho). \tag{69}$$

(b) Uniformly along sequences $\rho \to 0$ and $n\rho \to \infty$,

$$G_n(\rho) = (2 + o(1))\frac{\log(n\rho)}{\rho}. \tag{70}$$

More quantitatively, there are universal constants $C, L_0 < \infty$ such that, whenever $0 < \rho \le 1/4$ and $L_\rho := \log(n\rho) \ge L_0$,

$$\left|\frac{\rho G_n(\rho)}{2L_\rho} - 1\right| \le C\left(\rho + L_\rho^{-1/2} + \frac{\log L_\rho}{L_\rho}\right). \tag{71}$$

(c) For every fixed $c \ge 0$,

$$\frac{G_n(c/n)}{n} \longrightarrow 2\int_0^{1/2} \frac{dx}{1 + \lfloor cx \rfloor}. \tag{72}$$

Moreover, for every fixed $C_0 < \infty$, the convergence is uniform over $0 \le c \le C_0$, with error $O_{C_0}(1/n)$.

Proof. For part (a), first take $0 < \rho < 1$. Fix a large integer A. On the left half, whenever $u \ge A$,

$$(1 + \rho)u \le u + 1 + \lfloor \rho u \rfloor \le (1 + \rho + A^{-1})u. \tag{73}$$

The number of transitions before entering $[A, \infty)$ is $O(A)$. The first node above A and the first node above $n/2$ differ from their respective thresholds by bounded multiplicative factors. Taking logarithms in Eq. (73) therefore shows that the number of central left-half transitions, divided by $\log n$, lies between

$$\frac{1 + o_n(1)}{\log(1 + \rho + A^{-1})} \quad \text{and} \quad \frac{1 + o_n(1)}{\log(1 + \rho)}.$$

Letting first $n \to \infty$ and then $A \to \infty$ gives the first term in $\Psi(\rho)$.

On the right half put $y = n - u$. Before the capped terminal update,

$$y - 1 - \rho y \le y - 1 - \lfloor \rho y \rfloor \le (1 - \rho)y.$$

For $y \ge A$, this becomes

$$(1 - \rho - A^{-1})y \le y' \le (1 - \rho)y. \tag{74}$$

Choose $A > (1-\rho)^{-1}$. The same logarithmic argument, now run from a constant multiple of n down to A, gives $1/[-\log(1 - \rho)]$ after division by $\log n$. The midpoint-crossing overshoot is a bounded factor and the two endpoint pieces contribute $O(A)$ transitions. The node/transition convention contributes one.

If $\rho \ge 1$, every node $u \ge n/2$ has $u + 1 + \lfloor \rho(n - u) \rfloor \ge n + 1$, so the entire right phase is one transition. For $\rho > 1$, a left-half node becomes terminal as soon as $u + 1 + \lfloor \rho u \rfloor \ge n + 1$, which occurs at $u = n/(1 + \rho) + O(1)$; replacing $n/2$ by this constant multiple of n changes the transition count by only $O_\rho(1)$. The left argument therefore gives $1/\log(1 + \rho)$. This also covers $\rho = 1$, proving (a).

For part (b), put

$$L_\rho = \log(n\rho), \qquad \eta = L_\rho^{-1/2}, \qquad A = \left\lceil \frac{1}{\eta\rho} \right\rceil.$$

The hypotheses imply $L_\rho \to \infty$, $A = o(n)$, and, for $x \ge A$,

$$\rho x \le 1 + \lfloor \rho x \rfloor \le (1 + \eta)\rho x.$$

Applying the same logarithmic comparison on the increasing and decreasing halves, now with this A, gives

$$G_n(\rho) = \frac{2\log(n/A)}{\rho}\big(1 + O(\eta+\rho)\big) + O(A+1). \tag{75}$$

For completeness, the first entrance into the central left range overshoots A by at most a factor $1+O(\rho+\eta\rho)$; the midpoint crossing overshoots by $O(\rho n+1)$, leaving right distance $n/2+O(\rho n+1)$; and the first right distance below A differs from A by at most a factor $1 + O(\rho) + O(A^{-1})$. Their logarithms contribute $O(\rho + A^{-1})$, which is absorbed in Eq. (75). The left and right endpoint ranges contain at most $A + O(1)$ nodes each, because every update moves by at least one. Finally,

$$\log(n/A) = L_\rho + \log\eta + o(1) = L_\rho - \tfrac{1}{2}\log L_\rho + o(1) = (1+o(1))L_\rho$$

and

$$\frac{A}{L_\rho/\rho} = O\!\left(\frac{1}{\eta L_\rho}\right) = O(L_\rho^{-1/2}).$$

Dividing Eq. (75) by $2L_\rho/\rho$, using $\eta = L_\rho^{-1/2}$, $\log(n/A) = L_\rho - \frac{1}{2}\log L_\rho + O(1)$, and $\rho A/L_\rho = O(L_\rho^{-1/2})$, gives Eq. (71) after increasing universal C, L_0. Its right-hand side tends to zero under $\rho \to 0$ and $n\rho \to \infty$, proving (b).

For part (c), restrict c to a compact interval $0 \le c \le C_0$. On either half, an update at scaled distance $x = \min\{u/n, 1-u/n\}$ has integer length $h_c(x) = 1 + \lfloor cx \rfloor$. The interval $[0, 1/2]$ has at most $1 + \lfloor C_0/2 \rfloor$ subintervals on which h_c is a constant integer h. On a subinterval of length ℓ, a monotone trajectory with step h visits $n\ell/h + O(1)$ nodes; the boundary-crossing nodes contribute another $O(1)$, uniformly in c. Summing over the two halves gives

$$G_n(c/n) = 2n\int_0^{1/2} \frac{dx}{h_c(x)} + O_{C_0}(1).$$

The midpoint and terminal caps alter only $O_{C_0}(1)$ nodes. Dividing by n proves (c) and its locally uniform error bound. □

D.2 Inversion: sublogarithmic and logarithmic batches

Suppose first that $K = o(\log n)$. The optimal level tends to infinity: for every fixed ρ, Eq. (69) gives $G_n(\rho) \asymp \log n > K$. For $\rho \ge 1$, every uncapped update satisfies

$$F_\rho(u) \le (1+\rho)u + 1. \tag{76}$$

Iterating this affine upper bound from $u_1 = 1$ gives

$$u_{K+1} + \rho^{-1} \le (1+\rho)^K(1+\rho^{-1}). \tag{77}$$

Hence $G_n(\rho) \le K$ requires

$$(1+\rho)^K(1+\rho^{-1}) \ge n + 1 + \rho^{-1}. \tag{78}$$

Conversely, while the trajectory is in the left half, $1 + \lfloor \rho u \rfloor \ge \rho u$, so its uncapped update is at least $(1+\rho)u$. Once it enters the right half, the next update is terminal because $\rho \ge 1$. It follows that

$$(1+\rho)^K \ge n+1 \quad\Longrightarrow\quad G_n(\rho) \le K. \tag{79}$$

Apply Eqs. (78) and (79) at the threshold. Since $R_{n,K} \to \infty$,

$$1 + R_{n,K} = (1+o(1))n^{1/K}.$$

Also $n^{1/K} \to \infty$, so subtracting one proves part (i) of Theorem D.1.

Now assume $K/\log n \to \kappa \in (0,\infty)$. The function Ψ is continuous at 1, because the second term in its first branch tends to zero. Each branch has strictly negative derivative, its limits at zero and infinity are respectively infinity and zero, and hence there is a unique $r_\kappa > 0$ with $\Psi(r_\kappa) = \kappa$. For any sufficiently small fixed $\epsilon > 0$,

$$\Psi(r_\kappa - \epsilon) > \kappa > \Psi(r_\kappa + \epsilon).$$

By Eq. (69), the smaller level uses more than K nodes and the larger uses fewer than K nodes for all sufficiently large n. Greedy feasibility brackets $R_{n,K}$ between them. Letting $\epsilon \downarrow 0$ proves part (ii).

D.3 Inversion: mesoscopic and dense batches

Assume $K/\log n \to \infty$ and $K/n \to 0$, and put $L_K = \log(n/K)$. Then $L_K \to \infty$. For fixed $\epsilon \in (0,1)$, define

$$\rho_\pm = (2 \pm \epsilon)\frac{L_K}{K}.$$

We have $\rho_\pm \to 0$, $n\rho_\pm \to \infty$, and

$$\log(n\rho_\pm) = L_K + \log L_K + O(1) = (1 + o(1))L_K.$$

Part (b) of Lemma D.2 gives

$$\frac{G_n(\rho_-)}{K} \longrightarrow \frac{2}{2-\epsilon} > 1, \qquad \frac{G_n(\rho_+)}{K} \longrightarrow \frac{2}{2+\epsilon} < 1.$$

Greedy feasibility and then $\epsilon \downarrow 0$ prove part (iii). If $\log K/\log n \to \alpha < 1$, then $\log(n/K) = (1 - \alpha + o(1)) \log n$, proving Eq. (68).

For the dense regime, put $m = \lfloor c/2 \rfloor$. Directly partitioning $[0, 1/2]$ at the points j/c gives

$$\Phi(c) = \frac{2H_m}{c} + \frac{1 - 2m/c}{m+1}, \qquad 2m \le c < 2m+2, \tag{80}$$

where $H_m = \sum_{j=1}^{m} 1/j$. The adjacent formulas agree at $c = 2m+2$, so Φ is continuous. Inside each interval its derivative is

$$\Phi'(c) = -\frac{2}{c^2}\left(H_m - \frac{m}{m+1}\right) < 0.$$

Thus Φ is strictly decreasing, $\Phi(2) = 1$, and $\Phi(c) \to 0$ because $H_m = O(\log m)$. Every $\beta \in (0,1)$ therefore has a unique $c_\beta > 2$.

Choose $\epsilon > 0$ with $c_\beta - \epsilon > 2$. Part (c) gives

$$\frac{G_n((c_\beta - \epsilon)/n)}{n} \longrightarrow \Phi(c_\beta - \epsilon) > \beta,$$

and the analogous limit at $c_\beta + \epsilon$ is below β. If $K/n \to \beta$, greedy feasibility therefore yields

$$\frac{c_\beta - \epsilon}{n} < R_{n,K} \le \frac{c_\beta + \epsilon}{n}$$

eventually. Letting $\epsilon \downarrow 0$ proves part (iv).

It remains to treat $K/n \to 1$ with $K < n$. Every such schedule has a nonsingleton interval, so some $d_{\mathcal{N}}(i) \ge 1$. Since $w_i \le \lfloor n/2 \rfloor$,

$$R_{n,K} \ge \frac{1}{\lfloor n/2 \rfloor}. \tag{81}$$

Put $m = n - K = o(n)$. Starting from the all-singleton schedule, omit m mutually nonadjacent nodes in an interval of length $2m + O(1)$ centered at $n/2$. Every resulting nonsingleton interval has size two and a left endpoint u satisfying $w_u \ge n/2 - m - O(1)$. Thus

$$R_{n,K} \le \frac{1}{n/2 - m - O(1)}. \tag{82}$$

The two bounds give $nR_{n,K} \to 2$, proving part (v).

E Additional exact extensions

We collect the results used in the main text for larger alphabets, randomized cardinalities, hidden output, adaptive policies, and learned conditionals. Their proofs are in Appendix F.

Corollary E.1 (Large ambient alphabets)**.** If a finite token alphabet contains a prime-power subset of size at least $n - 1$, its deterministic minimax value is exactly $R_{n,K}$.

The upper bound is alphabet-free, while the matching projective-code targets can be supported entirely on that subset.

E.1 RANDOMIZED-CARDINALITY GAME

Theorem E.2 (Exact randomized-cardinality minimax)**.** Under the alphabet condition of Theorem 4.2, the two sides of Eq. (26) equal the labelled randomized minimax value. Both extrema are attained, and an optimal schedule law can be supported on at most n deterministic schedules.

E.2 HIDDEN OUTPUT AND ADAPTIVE POSITIONS

Let Θ be any finite schedule label with law μ, let q_θ be the corresponding exact-oracle output law, and put

$$\bar{q}_\mu(x) = \sum_\theta \mu(\theta) q_\theta(x), \qquad r_x(\theta) = \frac{\mu(\theta) q_\theta(x)}{\bar{q}_\mu(x)}. \tag{83}$$

Theorem E.3 (Hidden-label decomposition and deterministic minimax)**.** For every finite alphabet and schedule-label law,

$$\mathbb{E}_{\theta\sim\mu} D_{\mathrm{KL}}(p\|q_\theta) = D_{\mathrm{KL}}(p\|\bar{q}_\mu) + \mathbb{E}_{X\sim p} D_{\mathrm{KL}}(\mu\|r_X). \tag{84}$$

For every deterministic schedule and prime power $q \geq n-1$,

$$\sup_{p\neq\text{product}} \frac{\widetilde{E}_{\mathcal{N}}(p)}{\min\{\mathrm{TC}(p), \mathrm{DTC}(p)\}} = M_{\mathcal{N}}. \tag{85}$$

Thus hiding the coordinate order leaves the deterministic schedulewise minimax value unchanged.

Fix batch sizes $b_a = N_{a+1} - N_a$. A selection policy may choose each set of b_a unrevealed coordinates from the revealed transcript and exact one-coordinate conditionals, but cannot inspect unrevealed values. Let $P^{\mathrm{tr}}_{p,\mathcal{A}}$ and $Q^{\mathrm{tr}}_{p,\mathcal{A},\mathcal{N}}$ denote the target and product-sampler transcript laws, and let $Q^{\mathrm{out}}_{p,\mathcal{A},\mathcal{N}}$ be the latter's completed-vector marginal.

Theorem E.4 (Adaptive-coordinate minimax at fixed cardinalities)**.** For every prime power $q \geq n-1$, enlarging uniform coordinate orders to these policies leaves both deterministic minimax values equal to $R_{n,K}$:

$$\min_{\mathcal{N}} \inf_{\mathcal{A}} \sup_{p\neq\text{product}} \frac{D_{\mathrm{KL}}(P^{\mathrm{tr}}_{p,\mathcal{A}}\|Q^{\mathrm{tr}}_{p,\mathcal{A},\mathcal{N}})}{\min\{\mathrm{TC}(p), \mathrm{DTC}(p)\}} = R_{n,K}, \tag{86}$$

$$\min_{\mathcal{N}} \inf_{\mathcal{A}} \sup_{p\neq\text{product}} \frac{D_{\mathrm{KL}}(p\|Q^{\mathrm{out}}_{p,\mathcal{A},\mathcal{N}})}{\min\{\mathrm{TC}(p), \mathrm{DTC}(p)\}} = R_{n,K}. \tag{87}$$

E.3 TRANSCRIPT-ADAPTIVE CARDINALITIES

Let $\mathfrak{A}^{\mathrm{dyn}}_{n,K}$ contain policies that form exactly K nonempty batches. Before each batch, a policy may use private randomness, all previously revealed positions and values, and their exact one-coordinate conditionals to choose both the next batch size and its positions, but it may not inspect unrevealed values. Let $P^{\mathrm{tr}}_{p,\mathcal{A}}$ be its target transcript law and $Q^{\mathrm{tr}}_{p,\mathcal{A}}$ the corresponding exact-product sampler transcript law.

Theorem E.5 (Adaptive cardinalities do not improve labelled minimax)**.** For every prime power $q \geq n-1$,

$$\inf_{\mathcal{A}\in\mathfrak{A}^{\mathrm{dyn}}_{n,K}} \sup_{p\neq\text{product}} \frac{D_{\mathrm{KL}}(P^{\mathrm{tr}}_{p,\mathcal{A}}\|Q^{\mathrm{tr}}_{p,\mathcal{A}})}{\min\{\mathrm{TC}(p), \mathrm{DTC}(p)\}} = R^{\mathrm{lab}}_{n,K}. \tag{88}$$

Thus choosing later batch sizes and positions from the revealed transcript has the same labelled minimax value as sampling the complete schedule in advance.

E.4 LEARNED CONDITIONALS ACROSS SEQUENTIAL BLOCKS

For block r, let $H_r = (C, X^{(<r)})$. For a uniform size-s subset S of that block and uniform $i \notin S$, define the conditional predictor defect

$$\bar{\varepsilon}_{r,s} = \mathbb{E}_{H_r,S,i,X_S^{(r)}} D_{\mathrm{KL}}\Big(p_i(\cdot \mid H_r, X_S^{(r)}) \,\Big\|\, \widehat{p}_i(\cdot \mid H_r, X_S^{(r)})\Big). \tag{89}$$

If $b_{r,a}$ is batch a's size in block r, then

$$\widehat{E}^{\mathrm{lab}}_{\mathrm{hier}}(p) = E^{\mathrm{lab}}_{\mathrm{hier}}(p) + \sum_{r=1}^{B}\sum_{a=1}^{K_r} b_{r,a}\bar{\varepsilon}_{r,N_{r,a}-1}. \tag{90}$$

The learned hidden-output KL is at most this labelled quantity.

F Proofs for the additional exact extensions

F.1 Sequential-block composition

For side information C, define

$$\mathrm{TC}(X \mid C) = \sum_j H(X_j \mid C) - H(X \mid C), \tag{91}$$

$$\mathrm{DTC}(X \mid C) = H(X \mid C) - \sum_j H(X_j \mid X_{-j}, C). \tag{92}$$

Retaining every within-block order label and applying the KL chain rule gives

$$D(P^{\mathrm{lab}} \| Q^{\mathrm{lab}}) = \sum_{r=1}^{B} \mathbb{E}_{H_r} E_{\mathcal{N}_r}(p_{X^{(r)}|H_r}), \qquad H_r = (C, X^{(<r)}). \tag{93}$$

Applying Eq. (16) conditionally bounds this by $M_{\max}$ times the smaller of the summed local TC and DTC budgets. For TC, the block chain rule and conditioning reduces entropy give

$$\begin{aligned}\sum_r \mathbb{E}_{H_r}\,\mathrm{TC}(X^{(r)} \mid H_r) &= \sum_r \sum_{i\in r} H(X_i \mid C, X^{(<r)}) - H(X \mid C) \\ &\le \sum_i H(X_i \mid C) - H(X \mid C) = \mathrm{TC}(X \mid C).\end{aligned} \tag{94}$$

For DTC, additionally conditioning each coordinate on future blocks can only reduce its entropy, so

$$\begin{aligned}\sum_r \mathbb{E}_{H_r}\,\mathrm{DTC}(X^{(r)} \mid H_r) &= H(X \mid C) - \sum_r \sum_{i\in r} H(X_i \mid X^{(r)}_{-i}, C, X^{(<r)}) \\ &\le H(X \mid C) - \sum_i H(X_i \mid X_{-i}, C) = \mathrm{DTC}(X \mid C).\end{aligned} \tag{95}$$

This proves Eq. (24); data processing gives the hidden-output upper bound. For sharpness, place a projective Reed–Solomon single-jump law in a block attaining $M_{\max}$ and make all other coordinates independent. The global dependence budgets then reduce to that block, while the constant likelihood ratio in Eq. (99) attains the same loss with or without the local order label. This proves Eq. (25).

If only a total of K calls is fixed, where $B \le K \le \sum_r n_r$, the same greedy feasibility test gives

$$R^{\mathrm{hier}}_{\mathbf{n},K} = \min_{\substack{1\le K_r\le n_r \\ \sum_r K_r = K}} \max_r R_{n_r,K_r} = \inf\left\{\rho : \sum_{r=1}^{B} G_{n_r}(\rho) \le K\right\}. \tag{96}$$

Indeed, block r needs $G_{n_r}(\rho)$ calls at level ρ by Lemma 4.1; if their sum is below K, inserting unused boundaries cannot increase any load.

F.2 Randomized-cardinality minimax

Averaging Eq. (9) over $\mathcal{N} \sim \lambda$ yields

$$E_\lambda(p) = \sum_{i=1}^{n-1} \bar{d}_\lambda(i)\delta_i. \tag{97}$$

The coordinatewise upper bound is therefore $\max_i \bar{d}_\lambda(i)/w_i$. A single-jump code law at a maximizing index attains it, proving the primal side of Eq. (26). The payoff is bilinear on two finite simplices, so finite minimax duality gives the second side and attainment. Finally, $\bar{d}_\lambda \in \mathbb{R}^{n-1}$ lies in the convex hull of deterministic load vectors; Carathéodory elimination leaves at most n schedules in the support. This proves Theorem E.2.

F.3 HIDDEN-LABEL OUTPUT

For x with $p(x) > 0$, Eq. (83) gives

$$D_{\mathrm{KL}}(\mu \| r_x) = \log \bar{q}_\mu(x) - \mathbb{E}_{\theta\sim\mu} \log q_\theta(x). \tag{98}$$

Averaging under p and adding $D_{\mathrm{KL}}(p\|\bar{q}_\mu)$ proves Eq. (84).

For the deterministic minimax, fix the projective Reed–Solomon law p_k and the schedule interval $N_a \le k < N_{a+1}$. Every projection on at most k coordinates is uniform, while any k coordinates determine the codeword. Consequently, for every x in target support and every order π,

$$q_{\pi,\mathcal{N}}(x) = q^{-d_{\mathcal{N}}(k)} p_k(x). \tag{99}$$

The ratio is independent of x and π, so averaging the order gives output KL $d_{\mathcal{N}}(k) \log q$. Choosing a maximizing k matches the convexity upper bound $M_{\mathcal{N}}$, proving Eq. (85).

F.4 ADAPTIVE-COORDINATE CONVERSE

Fix a code witness p_k, a cardinality vector $\mathcal{N}$, and any policy in Theorem E.4. Before k coordinates are revealed, every unrevealed one-coordinate conditional is uniform; afterward, the codeword is determined. Thus the unique batch crossing k has sampler-to-target likelihood ratio $q^{-d_{\mathcal{N}}(k)}$, and all other batches have ratio one. The selection-policy kernels cancel because they are identical under the target and sampler transcript laws. The same constant ratio remains after marginalizing to the completed vector. Hence both transcript and output KL equal $d_{\mathcal{N}}(k) \log q$. Maximizing over k proves the lower bounds in Eqs. (86) and (87); uniform random order attains the upper bounds.

F.5 LEARNED-PREDICTOR DECOMPOSITIONS

Fix an order π and a position j in batch a. Let T be the preceding positions, S the prefix before batch a, and $i = \pi_j$. Add and subtract the exact conditional at the batch prefix:

$$\log \frac{p(X_i \mid X_T)}{\widehat{p}_i(X_i \mid X_S)} = \log \frac{p(X_i \mid X_T)}{p(X_i \mid X_S)} + \log \frac{p(X_i \mid X_S)}{\widehat{p}_i(X_i \mid X_S)}. \tag{100}$$

Taking expectations under p and averaging over uniform π, the first terms sum to $E_{\mathcal{N}}(p)$. Each of the b_a positions in batch a has the uniform (S, i) law in Eq. (33), so the second terms sum to $\sum_a b_a \varepsilon_{N_a-1}$. This proves Eq. (34); convexity gives the corresponding hidden-output upper bound.

Applying the same split conditionally on every outer history H_r and summing Eq. (93) proves Eq. (90). Uniform within-block order gives the (S, i) law in Eq. (89); data processing gives the learned hidden-output upper bound.

F.6 TRANSCRIPT-ADAPTIVE CARDINALITY CONVERSE

Fix a dynamic policy $\mathcal{A}$ and run a *uniform shadow execution*: feed it independent uniform field symbols and uniform one-coordinate conditionals for exactly K batches. The resulting random batch starts define a schedule law $\lambda_{\mathcal{A}}$. For an $[n, k]$ projective-code target, every transcript before the batch crossing k has the same law as this shadow execution by Lemma C.2. For the crossing batch a, the policy chooses N_{a+1} from a prefix of length $N_a - 1 < k$, which fixes $d_{\mathcal{N}}(k) = N_{a+1} - k - 1$. Hence

$$\mathbb{E}_{P^{\mathrm{tr}}_{p_k,\mathcal{A}}} d_{\mathcal{N}(\mathrm{tr})}(k) = \mathbb{E}_{\mathcal{N}\sim\lambda_{\mathcal{A}}} d_{\mathcal{N}}(k). \tag{101}$$

The policy kernels cancel between the target and sampler transcript laws, and the crossing-batch calculation gives

$$\frac{D_{\mathrm{KL}}(P^{\mathrm{tr}}_{p_k,\mathcal{A}} \| Q^{\mathrm{tr}}_{p_k,\mathcal{A}})}{w_k \log q} = \frac{\mathbb{E}_{\lambda_{\mathcal{A}}} d_{\mathcal{N}}(k)}{w_k}. \tag{102}$$

Maximizing over k and using Eq. (26) proves the lower bound $R^{\mathrm{lab}}_{n,K}$. Sampling an optimal schedule law in advance and then using a uniform order is an admissible dynamic policy and attains it, proving Theorem E.5.

G RANDOMIZED-LABELLED SHARP REGIMES

This appendix proves the new regimes summarized in Theorem 5.1. Throughout, the risk retains the sampled schedule label. The exact finite game itself is proved in Theorem E.2.

G.1 MESOSCOPIC LOWER BOUND

Put $L = \log(n/K)$, $A = \lceil K/L^{3/4} \rceil$, and

$$\mathcal{C}_A = \{i : A \le w_i \le \lfloor n/2 \rfloor\}, \qquad Z_A = \sum_{i \in \mathcal{C}_A} \frac{1}{w_i}. \tag{103}$$

Under $K/\log n \to \infty$ and $K/n \to 0$, we have $L \to \infty$, $A \to \infty$, $A = o(n)$, and $Z_A = (2+o(1))L$. Use the dual law

$$\gamma_i = \frac{\mathbf{1}\{i \in \mathcal{C}_A\}}{Z_A w_i}. \tag{104}$$

Define $\psi(h) = h + e^{-h} - 1$. Two elementary discrete estimates are

$$\sum_{i=a}^{q} \frac{q-i}{i^2} \ge \psi\left(\sum_{i=a}^{q-1} \frac{1}{i}\right), \tag{105}$$

$$\sum_{y=z}^{Y} \frac{y-z}{y^2} \ge \psi\left(\sum_{y=z+1}^{Y} \frac{1}{y}\right). \tag{106}$$

For completeness, the first follows by induction in q: the left increment is at least $h/q + q^{-2}$, while $\psi(h+q^{-1}) - \psi(h) \le h/q + (2q^2)^{-1}$ because $\psi'(h) \le h$. The second follows similarly from convexity and $e^{-\sum_{y=z+1}^{Y+1} 1/y} \ge z/(Y+1)$.

Intersect every schedule batch with the two halves of $\mathcal{C}_A$. This produces at most $J \le K+1$ nonempty pieces. If their harmonic lengths are $h_1, \dots, h_J$, at most one endpoint term per piece is lost, and hence

$$S := \sum_j h_j \ge Z_A - J/A = (2+o(1))L. \tag{107}$$

Equations (105) and (106), convexity, and the fact that $J\psi(S/J)$ decreases in J yield

$$\sum_{i \in \mathcal{C}_A} \frac{d_{\mathcal{N}}(i)}{w_i^2} \ge \sum_{j=1}^{J} \psi(h_j) \ge (K+1)\psi\left(\frac{S}{K+1}\right) \tag{108}$$

$$= (2+o(1))\frac{L^2}{K}. \tag{109}$$

Dividing by Z_A and applying the dual in Eq. (26) proves

$$R_{n,K}^{\mathrm{lab}} \ge (1-o(1))\frac{L}{K}. \tag{110}$$

G.2 A FINITE RANDOMIZED LOG-GRID CONSTRUCTION

Let $m = \lfloor n/2 \rfloor$, $c = m+1$, and $r = n+1-c$. Fix an integer $1 \le A < \min\{c, r\}$ and spacings $h_+, h_- > 0$. Sample independent phases $U \sim \mathrm{Unif}[0, h_+)$ and $V \sim \mathrm{Unif}[0, h_-)$, and use the start set

$$\begin{aligned} \mathcal{S}(U,V) = &\{1, \dots, A\} \cup \{n-A+1, \dots, n\} \cup \{c\} \\ &\cup \{\lceil Ae^{U+jh_+} \rceil : U + jh_+ < \log(c/A)\} \\ &\cup \{n+1-\lceil Ae^{V+jh_-} \rceil : V + jh_- < \log(r/A)\}. \end{aligned} \tag{111}$$

Here j ranges over the nonnegative integers in both grid sets. Duplicate starts are merged. Every realization has at most

$$2A + 1 + \left\lceil \frac{\log(c/A)}{h_+} \right\rceil + \left\lceil \frac{\log(r/A)}{h_-} \right\rceil \tag{112}$$

starts; if fewer than K remain, insert arbitrary unused starts. Such refinement only lowers every load coordinate.

Random phase gives the pointwise bounds

$$\frac{\mathbb{E}d(i)}{w_i} \le a_+(h_+) := \frac{e^{h_+}-1}{h_+} - 1 \qquad (i < c), \tag{113}$$

$$\frac{\mathbb{E}d(i)}{w_i} \le a_-(h_-) := 1 - \frac{1-e^{-h_-}}{h_-} \qquad (i \ge c). \tag{114}$$

Indeed, for $i < c$, the next unrounded left grid point has the form $Y = ie^Z$, where the residual phase Z is uniform on $(0, h_+]$. Rounding gives

$$d(i) \le \lceil Y \rceil - i - 1 \le Y - i = i(e^Z - 1).$$

For $i \ge c$, put $y = n - i = w_i$. The preceding grid point in reflected coordinates is $Y = ye^{-Z}$, with Z uniform on $[0, h_-)$. Its rounded start is $n + 1 - \lceil Y \rceil > i$, and

$$d(i) \le y - \lceil Y \rceil \le y - Y = y(1 - e^{-Z}).$$

These comparisons use the phase grids extended beyond their cutoffs. The deterministic tails and center supply next starts no later than these comparisons, and refinement only decreases loads. Integrating over Z gives Eqs. (113) and (114), including the central indices for either parity of n.

Choose $A = \lceil K/L^{3/4} \rceil$, reserve the constant ceiling overhead in Eq. (112), and allocate the remaining grid points proportionally to $\log(c/A)$ and $\log(r/A)$. Then $h_+ = h_- = (2 + o(1))L/K = o(1)$ and every realization uses at most K starts. Since $a_+(h) = h/2 + O(h^2)$ and $a_-(h) = h/2 + O(h^2)$,

$$R^{\mathrm{lab}}_{n,K} \le (1 + o(1))\frac{L}{K}. \tag{115}$$

Together with Eq. (110), this proves Eq. (27).

G.3 Dense exact family

Let $K = n - m$, $0 \le m \le \lfloor n/2 \rfloor$, and $W_n = \sum_i w_i = \lfloor n^2/4 \rfloor$. The balanced dual $\gamma_i = w_i/W_n$ gives

$$R^{\mathrm{lab}}_{n,K} \ge \frac{m}{W_n}, \tag{116}$$

because for every K-batch schedule with batch sizes b_a,

$$\sum_i d_{\mathcal{N}}(i) = \sum_a \binom{b_a}{2} \ge \sum_a (b_a - 1) = m. \tag{117}$$

Suppose $m(n-1) \le W_n$ and set $p_i = mw_i/W_n$. Then $\sum_i p_i = m$ and $p_i + p_{i+1} \le 1$. Partition $[0, m)$ into consecutive half-open intervals of lengths p_i, sample $U \sim \mathrm{Unif}[0, 1)$, and select the interval indices containing $U, U + 1, \ldots, U + m - 1$. Exactly m distinct, nonadjacent indices are selected and index i has marginal probability p_i. Omitting start $i + 1$ for each selected index produces only singleton and pair batches, with

$$\mathbb{E}d_{\mathcal{N}}(i) = p_i = \frac{mw_i}{W_n}. \tag{118}$$

This matches Eq. (116).

Conversely, equality in Eq. (116) forces equality in Eq. (117) almost surely, so every supported schedule uses only singletons and pairs and $\mathbb{E}d(i) = mw_i/W_n$ at every index. Adjacent omitted starts cannot coexist; hence

$$1 \ge \mathbb{E}[d(i) + d(i+1)] = \frac{m(w_i + w_{i+1})}{W_n}. \tag{119}$$

The maximum adjacent weight sum is $n - 1$. Therefore equality is possible iff $m(n-1) \le W_n$, equivalently $m \le \lfloor (n+1)/4 \rfloor$. This proves Eq. (28).

For the boundary $K/n \to 3/4$, define

$$C_n(\rho) = \sum_i \min\{\rho w_i, 1/2\}, \qquad \rho^{\mathrm{pair}}_{n,m} = \inf\{\rho : C_n(\rho) \geq m\}. \tag{120}$$

The same systematic sampler, now with capped marginals, gives $R^{\mathrm{lab}}_{n,K} \leq \rho^{\mathrm{pair}}_{n,m}$. If $m/n \to \delta < 1/2$, a Riemann sum gives

$$n\rho^{\mathrm{pair}}_{n,m} \to \begin{cases} 4\delta, & 0 \leq \delta \leq 1/4, \\ [4(1/2-\delta)]^{-1}, & 1/4 \leq \delta < 1/2. \end{cases} \tag{121}$$

The first branch and Eq. (116) close every sequence with $K/n \to \beta \geq 3/4$.

G.4 Two batches

A two-batch schedule is determined by its first-batch size S. Define

$$A_M = \sum_{i=1}^{M-1} \frac{1}{n-i}, \qquad M_n = \max\{M : A_M \leq M/n\}. \tag{122}$$

For $n \geq 8$, put $M = M_n$ and

$$\rho_n = \frac{A_M(n-M)}{M - A_M(n-M)}. \tag{123}$$

Then $R^{\mathrm{lab}}_{n,K} = \rho_n$. A matching primal distribution is

$$\Pr(S=i) = \frac{1+\rho_n}{n-i}\ (i < M), \qquad \Pr(S=M) = 1 - (1+\rho_n)A_M. \tag{124}$$

The definition of M makes the last mass nonnegative. If $y_i = \mathbb{E}d_S(i)$, the identity $y_{i+1} - y_i = -1 + (n-i)\Pr(S=i)$ gives $y_i = \rho_n i$ for $i \leq M$ and $y_i = n - i$ thereafter; maximality of M checks all remaining normalized-load constraints.

For the dual, take $\gamma_i = 0$ for $i > M$ and recursively choose

$$\gamma_{s+1} = \frac{s+1}{n-s-1} \sum_{i=1}^{s} \frac{\gamma_i}{i}, \qquad s < M, \tag{125}$$

then normalize. Consecutive schedule costs are equal for $S \leq M$ and strictly increase afterward, so averaging against Eq. (124) gives the matching value Eq. (123).

Finally $A_M \leq M/n < A_{M+1}$ implies $M \sim \sqrt{2n}$, $nA_M - M = o(1)$, and $MA_M \to 2$. Substitution in Eq. (123) yields

$$R^{\mathrm{lab}}_{n,K} \sim \sqrt{n/2}. \tag{126}$$

H The nonlinear actual-output game

All logarithms in this appendix are natural. Let A denote widths $(1,2)$ and B widths $(2,1)$ for $(n,K) = (3,2)$. If k is the singleton coordinate of A and the last coordinate of B, with $\{i,j,k\} = [3]$, their order-specific output laws on target-positive cells are

$$q_{A,k}(x) = \frac{p_{ik}(x_i,x_k)p_{jk}(x_j,x_k)}{p_k(x_k)}, \tag{127}$$

$$q_{B,k}(x) = p(x)\frac{p_i(x_i)p_j(x_j)}{p_{ij}(x_i,x_j)}. \tag{128}$$

Thus $Q_A = \frac{1}{3}\sum_k q_{A,k}$ and $Q_B = \frac{1}{3}\sum_k q_{B,k}$. Conditionals at null prefixes may be chosen arbitrarily because they do not affect forward KL.

H.1 ALPHABET-FREE UPPER BOUND

Define

$$P_2 = \sum_{i<j} I(X_i; X_j), \qquad C_2 = \sum_{k=1}^{3} I(X_i; X_j \mid X_k), \tag{129}$$

where $\{i, j\} = [3] \setminus \{k\}$ in the second sum. Entropy expansion gives

$$P_2 = 2\,\mathrm{TC} - \mathrm{DTC}, \qquad C_2 = 2\,\mathrm{DTC} - \mathrm{TC}, \tag{130}$$

and hence $\mathrm{TC}/2 \le \mathrm{DTC} \le 2\,\mathrm{TC}$. The expected labelled KL for the law that chooses A with probability t is

$$\begin{aligned} L_t(p) &= \frac{t}{3} C_2 + \frac{1-t}{3} P_2 \\ &= \left(\frac{2}{3} - t\right) \mathrm{TC} + \left(t - \frac{1}{3}\right) \mathrm{DTC}\,. \end{aligned} \tag{131}$$

Convexity of KL gives $D_{\mathrm{KL}}(p\|q_t) \le L_t(p)$. Therefore

$$D_{\mathrm{KL}}(p\|q_t) \le t\,\mathrm{TC} \quad (t \ge 1/3), \qquad D_{\mathrm{KL}}(p\|q_t) \le (1-t)\,\mathrm{DTC} \quad (t \le 2/3). \tag{132}$$

For $t \in [1/2, 2/3]$, these bounds are at most $t\,\mathrm{TC}$ and $t\,\mathrm{DTC}$. If $t \ge 2/3$, the TC coefficient in Eq. (131) is nonpositive, so $L_t \le (t - 1/3)\,\mathrm{DTC} \le t\,\mathrm{DTC}$; the first bound still gives $L_t \le t\,\mathrm{TC}$. Hence $D_{\mathrm{KL}}(p\|q_t) \le t \min\{\mathrm{TC}, \mathrm{DTC}\}$ for every $t \ge 1/2$.

H.2 MATCHING SUPPORT-THINNED TARGETS

Let π_ϵ be the product law of three independent $\mathrm{Bernoulli}(\epsilon)$ variables and define

$$p_\epsilon(x) = \pi_\epsilon(x) + \epsilon^3 (-1)^{|x|}, \qquad 0 < \epsilon < 1/2. \tag{133}$$

The cell 111 has zero mass and all other cells are nonnegative. Summing the parity term over any omitted coordinate gives zero, so every proper marginal of p_ϵ equals that of π_ϵ. Thus all three pairs are independent and

$$Q_A(p_\epsilon) = \pi_\epsilon, \qquad Q_B(p_\epsilon) = p_\epsilon, \qquad q_t = (1-t)p_\epsilon + t\pi_\epsilon. \tag{134}$$

Pairwise independence gives $\mathrm{DTC}(p_\epsilon) = 2\,\mathrm{TC}(p_\epsilon)$, while a Taylor expansion on the seven positive cells gives

$$\mathrm{TC}(p_\epsilon) = \epsilon^3 + O(\epsilon^4), \qquad D_{\mathrm{KL}}(p_\epsilon\|q_t) = t\epsilon^3 + O(\epsilon^4). \tag{135}$$

Hence $V_{\mathcal{X}}(t) \ge t$ for every $t > 0$ and every alphabet with at least two symbols. Together with Eq. (132), this proves Eq. (31).

H.3 A SECOND EXTREMAL FAMILY AND THE OPTIMIZED BRACKET

Consider the common-bit family $U \sim \mathrm{Bernoulli}(r)$ and $X_1 = X_2 = X_3 = U$. Write $H_b(r) = -r \log r - (1-r)\log(1-r)$ for binary entropy. Then

$$\mathrm{TC} = 2H_b(r), \qquad \mathrm{DTC} = H_b(r), \tag{136}$$

and direct evaluation gives

$$K_t(r) = D_{\mathrm{KL}}(p\|q_t) = -r \log(t + (1-t)r) - (1-r)\log(t + (1-t)(1-r)). \tag{137}$$

Moreover,

$$K_t(r) \le h(t) H_b(r), \qquad h(t) = \frac{-\log((1+t)/2)}{\log 2}, \tag{138}$$

with equality at $r = 1/2$. The cases $t = 0, 1$ are immediate. For $0 < t < 1$, define the increasing, concave function

$$\phi_t(x) = 1 - \left(\frac{t}{t + (1-t)x}\right)^2.$$

Direct differentiation gives

$$\frac{K_t''(r)}{H_b''(r)} = (1-r)\phi_t(r) + r\phi_t(1-r) =: g_t(r).$$

For $0 < r < 1/2$, monotonicity and concavity of ϕ_t imply

$$g_t'(r) = \phi_t(1-r) - \phi_t(r) + (1-r)\phi_t'(r) - r\phi_t'(1-r) \geq 0.$$

Since $K_t'(1/2) = H_b'(1/2) = 0$, the ratio $K_t'(r)/H_b'(r)$ is the average of g_t on $[r, 1/2]$ with positive weight $-H_b''$. This average increases with r. As H_b also increases on $[0, 1/2]$, K_t is convex as a function of H_b there. The chord from $(0, 0)$ to $(\log 2, -\log((1+t)/2))$ proves Eq. (138) on this interval; symmetry under $r \mapsto 1-r$ gives the remaining cases.

The support-thinned and uniform common-bit families give

$$V_2(t) \geq \max\{t, h(t)\}. \tag{139}$$

The two terms meet once, at $\rho \in (0, 1/2)$ satisfying $2^{1-\rho} = 1 + \rho$. Hence $\inf_t V_2(t) \geq \rho$. Choosing $t = 1/2$ in Eq. (31) gives the upper bound $1/2$, proving Eq. (32).

I Pareto geometry and scalable certificates

I.1 Sharp coefficient pairs

For a schedule law λ, let

$$A_\lambda = \max_i \frac{\bar{d}_\lambda(i)}{n-i}, \qquad B_\lambda = \max_i \frac{\bar{d}_\lambda(i)}{i}. \tag{140}$$

The jump identity and nonnegativity give $E_\lambda(p) \leq A_\lambda \,\mathrm{TC}(p)$ and $E_\lambda(p) \leq B_\lambda \,\mathrm{DTC}(p)$. Under the projective-code alphabet condition, a single-jump law at a maximizing coordinate attains each ratio. Thus these are the sharp schedulewise coefficients. If $\mathcal{R}_{n,K}$ is the upper region of achievable randomized-labelled pairs, then

$$R_{n,K}^{\mathrm{lab}} = \inf\{C : (C, C) \in \mathcal{R}_{n,K}\}. \tag{141}$$

This proves the diagonal-contact interpretation used in Section 3.

I.2 Complete near-sequential Pareto frontier

When $K = n-1$, every schedule omits exactly one internal start. Index the schedule by that omitted information boundary i. Its load vector is the ith coordinate vector, so a mixture has $\bar{d}(i) = \lambda_i$. Define

$$S_n(A, B) = \sum_{i=1}^{n-1} \min\{A(n-i), Bi\}. \tag{142}$$

Theorem I.1 (Randomized-labelled frontier for $K = n-1$)**.** The upper region is

$$\mathcal{R}_{n,n-1} = \{(A, B) : S_n(A, B) \geq 1\}. \tag{143}$$

Its nondominated frontier is $S_n(A, B) = 1$ with $1/(n-1) \leq A/B \leq n-1$. At every such point the unique schedule law is

$$\lambda_i = \min\{A(n-i), Bi\}. \tag{144}$$

The frontier has vertices

$$V_m = \left(\frac{2}{n(n-m)}, \frac{2}{nm}\right), \qquad m = 1, \ldots, n-1. \tag{145}$$

Proof. The pair (A, B) is feasible iff a probability vector λ satisfies $\lambda_i \leq A(n-i)$ and $\lambda_i \leq Bi$ for every i. Such a vector exists iff the sum of these coordinate capacities is at least one, proving Eq. (143). On the boundary every capacity must be saturated, giving uniqueness and Eq. (144). Outside the stated ratio interval, one coordinate bound is redundant and the point is dominated. Inside it, let m be the last index for which $Bi \leq A(n-i)$. Summation gives

$$\frac{m(m+1)}{2} B + \frac{(n-m-1)(n-m)}{2} A = 1. \tag{146}$$

Adjacent segments meet at Eq. (145). □

Every nonzero nonnegative linear preference satisfies

$$\min_{(A,B)\in\mathcal{R}_{n,n-1}}(\alpha A+\beta B)=\frac{2}{n}\min_{1\le m<n}\left(\frac{\alpha}{n-m}+\frac{\beta}{m}\right). \tag{147}$$

Thus randomization contracts every linear pair scalarization by $2/n$ while preserving its optimal pivots. The common-maximum criterion is different: its improvement factor is $\lceil n/2\rceil$, because it asks where the region first touches the diagonal. This distinction is why it is misleading to convexify already-maximized coefficient pairs; randomization convexifies load vectors before either maximum is taken.

I.3 Exact deterministic recovery

Let $Q=\lfloor n/2\rfloor$. For any deterministic schedule,

$$M_{\mathcal{N}}=\max_{a:N_a<n}\frac{N_{a+1}-N_a-1}{w_{N_a}}, \tag{148}$$

so its reduced denominator is at most Q. The optimum, being the objective of an attaining schedule, has the same bound. Two distinct reduced rationals with denominators at most Q differ by at least Q^{-2}.

For $K=n$, return singleton batches and value zero in $O(K)$ operations. For $K<n$, start with the exact bracket $(L,H]=(0,n-K]$, where L is infeasible and H feasible. Test dyadic midpoints with Eq. (17). After

$$T=\mathrm{bitlength}((n-K)Q^2)=O(\log n) \tag{149}$$

tests, $H-L<Q^{-2}$. The exact objective of the final feasible greedy schedule and $R_{n,K}$ both lie in this interval and both have denominator at most Q; hence they coincide. Each test visits at most $K+1$ nodes. This proves the $O(K\log n)$ arithmetic and $O(K)$ memory bounds in Theorem 4.4. Stopping when $H-L\le\epsilon$ instead returns an actual schedule objective U and certificate

$$L<R_{n,K}\le U\le H,\qquad U-R_{n,K}\le\epsilon. \tag{150}$$

I.4 A linear-time-per-layer randomized best response

For a fixed dual vector in Eq. (26), put $y_i=\gamma_i/w_i\ge 0$ and $y_n=0$. A schedule edge $u\to v$ has cost

$$C_y(u,v)=\sum_{i=u}^{v-1}(v-i-1)y_i. \tag{151}$$

Its adjacent cross-difference is

$$C_y(u,v)+C_y(u+1,v+1)-C_y(u,v+1)-C_y(u+1,v)=-y_u\le 0. \tag{152}$$

With prefix sums $S_0(j)=\sum_{i\le j}y_i$ and $S_1(j)=\sum_{i\le j}iy_i$,

$$C_y(u,v)=A(v)+S_1(u-1)-(v-1)S_0(u-1), \tag{153}$$

where $A(v)=(v-1)S_0(v-1)-S_1(v-1)$ and $S_0(0)=S_1(0)=0$. Let $D_a(v)$ be the minimum cost of a path from 1 to v with exactly a edges. Set $D_0(1)=0$ and $D_0(v)=+\infty$ for $v>1$. For $a=1,\dots,K$, the recurrence is

$$D_a(v)=\min_{1\le u<v}\{D_{a-1}(u)+C_y(u,v)\},\qquad D_a(1)=+\infty.$$

The best-response value is $D_K(n+1)$. At layer a, each reachable predecessor u contributes the line

$$\ell_u(x)=-S_0(u-1)x+D_{a-1}(u)+S_1(u-1).$$

Process $v=2,\dots,n+1$ in order. Before querying at $x=v-1$, insert the line for $u=v-1$ if it is reachable in the previous layer. The query then minimizes over exactly the reachable predecessors $u<v$; adding $A(v)$ gives $D_a(v)$. The slopes are nonincreasing because $y_i\ge 0$, and query points increase. A deque lower hull evaluates each of the K layers in $O(n)$ arithmetic operations, giving an $O(Kn)$ exact best response.

This oracle yields a global lower bound for any restricted primal and hence an exact gap certificate for column generation. It also supplies a standard polynomial-time separation oracle for the finite LP. An explicit primal law can be compressed to at most n schedules while preserving all $n-1$ expected-load coordinates, by affine dependence and Carathéodory elimination.

J EXPERIMENTAL PROTOCOLS AND SUPPORTING RESULTS

J.1 UNCERTAINTY AND DATA SPLITS

Error bars on GSM8K results are 95% intervals over prompts. We checked the schedule solvers against exhaustive enumeration on small instances. The seven-call configuration and the entropy threshold were chosen on development splits of the GSM8K training set; the seven-call comparison, the five-budget curve, and the shape comparison use disjoint GSM8K test prompts.

J.2 PROTECTION UNDER PROFILE SHIFT

At $n = 8$, we interpolate joint laws as

$$p_{\alpha,f} = (1-\alpha)p_{\mathrm{cal}} + \alpha p_f, \qquad \alpha \in \{0, 1/4, 1/2, 3/4, 1\}.$$

The calibration law mixes, with weights 0.55 and 0.45, a product of two independent latent blocks of four bits with flip probabilities 0.12 and 0.22, and a symmetric binary Markov chain with stay probability 0.78. Each latent block copies a fair bit with independent flips; each Markov chain starts with a fair bit. The first two endpoints p_f are copies of one fair latent bit with flip probability 0.035, and a mixture of the uniform even-parity law and the uniform law on $\{0,1\}^8$ with weights 0.97 and 0.03. The other endpoints use two independent latent blocks of four bits with flip probabilities 0.025 and 0.18, and a symmetric Markov chain with stay probability 0.94. For each $K \in \{3,4,5\}$, method, and shift level, we compute the normalized labelled risk under each $p_{\alpha,f}$ exactly and plot its maximum over f.

At zero shift, the deterministic robust schedule exceeds the evaluation oracle by 0.067, 0.094, and 0.067 for $K = 3, 4, 5$. The full curves in Figure 2 show the worst normalized labelled risks across all five shift levels.

J.3 LEARNED CONDITIONALS AND TARGET-AWARE SELECTION

This experiment uses length-eight binary sequences, 12 target laws from common-bit, parity, latent-block, Ising-cycle, and Markov families, and three training seeds. For each target, a masked predictor is trained from 8,192 sequences and 65,536 pairs of masks and queries. At $K = 4$, we compare equal widths $[2,2,2,2]$, the robust schedule $[1,2,4,1]$, and a target-aware comparator selected from all 35 schedules with four nonempty batches. For each target, this comparator minimizes oracle labelled KL. The finite binary laws allow dynamic programming to enumerate the complete output distribution with the schedule hidden. The maximum predictor defect is 0.069 nats.

This experiment and the confirmation experiment below use the same training sample sizes, predictor, and training procedure. The input concatenates one-hot encodings of each position's masked, zero, or one state with a one-hot query position. A network with two hidden layers of 64 units and GELU activations produces a logit. A learned scalar for each query position and observed context is added to this logit and initialized to zero. We train both components by binary cross-entropy with AdamW for 5,000 steps, using batches of 512 queries, learning rate 0.002, and weight decay 0.0001.

Training sequences are independent draws from the target law. Each training query samples one of these sequences with replacement, a uniform query position, and a uniform observed count in $\{0,\dots,7\}$. The observed subset is uniform among the other positions of that size. Training batches sample the resulting queries with replacement. At evaluation, sigmoid probabilities are clipped to $[10^{-6}, 1-10^{-6}]$.

Mean output KL is 0.121, 0.069, and 0.050 for equal, robust, and target-aware schedules (Figure 4). Their mean labelled KL values are 0.436, 0.413, and 0.349.

To test whether oracle output risk predicts schedule ordering under learned conditionals, a separate confirmation bank uses 12 new positive binary laws, all 35 schedules with four nonempty batches, and three new predictor seeds. Output laws are again enumerated exactly. For each schedule, each KL diagnostic is the maximum normalized risk over target laws and, for learned predictors, over seeds. Rankings and Spearman associations compare the 35 schedules (Table 1). The maximum predictor defect is 0.065 nats. Panel A contains the two heterogeneous latent laws, both unequal block laws,

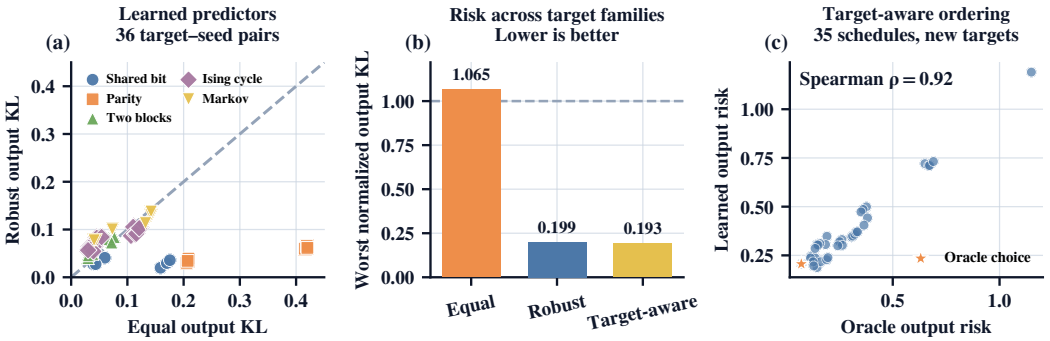

Figure 4: **Learned conditionals and target-aware selection.** (a) Paired hidden-output KL over 36 combinations of targets and predictors. (b) Worst normalized hidden-output risk for equal, robust, and target-aware schedules. (c) Across all 35 schedules in the prespecified confirmation bank, oracle output risk predicts learned-output ordering; the star marks the oracle-selected schedule. All output probabilities are evaluated by exact enumeration.

the star, and the ladder. Panel B contains the other two graphical laws, both asymmetric Markov laws, and both higher-order laws. The target distributions are specified below.

Table 1: Spearman association with worst learned hidden-output KL on the prespecified confirmation bank.

Schedule diagnostic	Global	Panel A	Panel B
Oracle hidden-output KL	0.9171	0.9936	0.8739
Learned labelled KL	0.7863	0.6527	0.5908
Oracle factorization KL	0.7810	0.9532	0.8513
Distribution-free load	0.4611	0.7990	−0.2377

The oracle-output diagnostic selects widths $[1, 3, 2, 2]$, ranking third of 35 under learned output risk. Equal widths rank first on this bank, whereas the robust schedule ranks 19th. These rankings support using oracle output risk to select schedules for this target family under learned conditionals.

Target distributions. All laws are on $\{0,1\}^8$. We index coordinates from 1 to 8 and write $z_i = 2x_i - 1$. A noisy latent block copies a bit $H \sim \text{Bernoulli}(w)$, flipping coordinate i independently with probability η_i. Different blocks have independent latent bits and flips. Exponential formulas below are normalized over all 2^8 states. Table 2 specifies the 12 laws of the learned-conditional experiment.

Table 2: Target laws for the learned-conditional experiment. Each listed parameter choice defines one law.

Family	Definition and parameters
Common latent bit	$w = 1/2$, all $\eta_i = \eta$, with $\eta \in \{0.10, 0.20\}$.
Soft parity	$p(x) \propto \exp(\beta \prod_{i=1}^{8} z_i)$, $\beta \in \{0.7, 1.2\}$.
Independent blocks	Blocks $\{1,2,3,4\}$ and $\{5,6,7,8\}$, both with $w = 1/2$ and the same flip probability $\eta \in \{0.10, 0.20\}$.
Ising cycle	$p(x) \propto \exp(J \sum_{i=1}^{8} z_i z_{i+1} + 0.08 \sum_{i=1}^{8} z_i)$, $z_9 = z_1$, $J \in \{0.45, 0.80, -0.45, -0.80\}$.
Markov chain	X_1 is fair; each transition flips the previous bit with probability $\eta \in \{0.10, 0.25\}$, the same in both directions.

The confirmation bank consists of the eight laws in Table 3 and the four graphical laws in Table 4. In the Markov rows, $\pi = \Pr(X_1 = 1)$, $u_i = \Pr(X_{i+1} = 1 \mid X_i = 0)$, and $v_i = \Pr(X_{i+1} = 0 \mid X_i = 1)$ for $i = 1, \ldots, 7$. In the higher-order rows, $p(x) \propto \exp(0.04 \sum_i z_i + f(z))$. The graphical laws

have

$$p(x) \propto \exp\left(\sum_{i=1}^{8} h_i z_i + \sum_{(i,j)\in E} J_{ij} z_i z_j\right).$$

Only the listed edges have nonzero couplings.

Table 3: Latent, Markov, and higher-order laws in the confirmation bank. All parameter vectors follow increasing coordinate order.

Law	Parameters
Common latent bit	$w = 0.37$, $\eta = (0.04, 0.08, 0.13, 0.19, 0.07, 0.16, 0.22, 0.11)$.
Common latent bit	$w = 0.63$, $\eta = (0.18, 0.06, 0.24, 0.09, 0.14, 0.21, 0.05, 0.12)$.
Blocks of sizes $3, 5$	Consecutive blocks, $w = 1/2$ in each; flip probabilities $0.09, 0.17$, respectively.
Blocks of sizes $2, 3, 3$	Consecutive blocks, $w = 1/2$ in each; flip probabilities $0.07, 0.14, 0.21$, respectively.
Markov chain	$\pi = 0.31$, $u = (0.08, 0.13, 0.19, 0.11, 0.23, 0.09, 0.16)$, $v = (0.27, 0.18, 0.12, 0.24, 0.15, 0.29, 0.20)$.
Markov chain	$\pi = 0.68$, $u = (0.21, 0.07, 0.16, 0.25, 0.10, 0.18, 0.12)$, $v = (0.09, 0.22, 0.14, 0.11, 0.26, 0.17, 0.23)$.
Higher-order law	$f(z) = 0.72 z_1 z_2 z_3 - 0.54 z_3 z_4 z_5$ $+ 0.43 z_5 z_6 z_7 z_8 + 0.31 z_1 z_6 z_8$.
Higher-order law	$f(z) = -0.64 z_1 z_3 z_6 + 0.58 z_2 z_4 z_7$ $- 0.39 z_3 z_5 z_7 z_8 + 0.47 z_1 z_2 z_5 z_8$.

Table 4: Graphical laws in the confirmation bank. Edge triples give (i, j, J_{ij}); field vectors follow coordinate order.

Graph	Couplings and fields
Star	$E = \{(1, j) : j = 2, \ldots, 8\}$, $(J_{12}, \ldots, J_{18}) = (0.58, 0.54, 0.50, 0.46, 0.42, 0.38, 0.34)$, $h = (0.09, -0.04, 0.03, -0.08, 0.06, -0.02, 0.04, -0.05)$.
Ladder	$(1, 2, 0.48), (3, 4, 0.52), (5, 6, 0.44), (7, 8, 0.50)$, $(1, 3, 0.37), (3, 5, 0.41), (5, 7, 0.35)$, $(2, 4, 0.33), (4, 6, 0.39), (6, 8, 0.36)$, $h = (0.03, -0.04, 0.05, -0.03, 0.02, -0.06, 0.04, -0.01)$.
Mixed signs	$(1, 3, 0.61), (2, 5, -0.53), (3, 6, 0.47)$, $(4, 8, -0.58), (1, 7, 0.34), (5, 8, 0.42)$, $h = (-0.07, 0.05, 0.02, -0.04, 0.08, -0.02, 0.06, -0.05)$.
Frustrated ring	$J_{i,i+1} = 0.46$ for $i = 1, \ldots, 7$, $J_{8,1} = -0.43$, $h = (0.04, -0.03, 0.06, -0.05, 0.02, -0.04, 0.05, -0.02)$.

J.4 GSM8K PROTOCOL AND FIVE-BUDGET RESULTS

We evaluate LLaDA-8B-Instruct (Nie et al., 2025) and Dream-7B-Instruct (Ye et al., 2025) in bfloat16 on one RTX 3090. Both use GSM8K (Cobbe et al., 2021), 128 generated tokens, greedy no-remasking, and the same answer extractor. Paired conditions share prompts, tokenization, temperature, output length, and nominal call budget. Each LLaDA configuration in Table 6 uses sequential blocks of the reported size with K calls per block. Equal and robust cuts share a frozen DUS-derived position priority within each configuration and differ only in batch boundaries. The complete-system reference is the official base-2 DUS no-remasking scheduler (Luxembourg et al., 2026), which chooses both position groups and cuts.

Within each block of length b, index positions by $0, \ldots, b-1$. For strides $b/2, b/4, \ldots, 1$, append unused indices divisible by the stride in increasing order. This list is the frozen priority for both methods; each call takes the next group of the size in Table 5. Blocks are processed sequentially.

Table 5: Per-block batch sizes for the five held-out configurations. Both methods use the same DUS-derived position priority within each block.

Block	K	Equal	Robust
128	7	$19, 19, 18, 18, 18, 18, 18$	$2, 4, 8, 16, 32, 64, 2$
64	6	$11, 11, 11, 11, 10, 10$	$2, 4, 8, 16, 32, 2$
32	5	$7, 7, 6, 6, 6$	$2, 4, 8, 16, 2$
16	4	$4, 4, 4, 4$	$2, 4, 8, 2$
8	3	$3, 3, 2$	$2, 4, 2$

Table 6: Accuracy (%) at the five budgets of Figure 3a under a fixed DUS-derived ranking. Difference is robust minus equal, in percentage points.

Block	K	NFE	Equal	Robust	DUS	Difference
128	7	7	35.55	47.66	54.69	$+12.11$
64	6	12	41.02	57.81	60.16	$+16.80$
32	5	20	45.31	69.92	67.58	$+24.61$
16	4	32	61.33	70.31	71.88	$+8.98$
8	3	48	69.14	74.22	75.00	$+5.08$

J.5 LLaDA and Dream at 32 calls

At 32 calls under uniform position selection, LLaDA generates four 32-token blocks with eight calls per block, while Dream generates one 128-token block with 32 calls. The paired conditions differ only in their batch boundaries (Table 7).

Table 7: Uniform-position comparison at 32 calls. Time is mean seconds per prompt; memory is the peak allocated GPU memory.

Model	Cuts	Exact match (%)	Seconds	Tokens/s	GiB
LLaDA	Equal	46.88	2.770	46.22	15.01
	Robust	**61.72**	2.768	46.24	15.01
Dream	Equal	26.56	2.284	56.05	14.40
	Robust	**38.28**	2.280	56.13	14.40

LLaDA uses 256 prompts and gives a paired robust-minus-equal gain of $+14.84$ points; Dream uses 128 prompts and gives $+11.72$ points. Runtime and memory remain nearly identical within each model.

J.6 Shape and dynamic-information controls

On 128 prompts and three coordinate seeds, all shape controls generate four sequential 32-token blocks with eight calls per block, totaling 32 calls. The robust per-block vector is $[1, 2, 3, 5, 9, 8, 3, 1]$; alternatives are equal $[4, 4, 4, 4, 4, 4, 4, 4]$, early-heavy $[1, 5, 5, 5, 4, 4, 4, 4]$, late-heavy $[4, 4, 4, 4, 5, 5, 5, 1]$, ends-heavy $[1, 5, 5, 5, 5, 5, 5, 1]$, reversed $[1, 3, 8, 9, 5, 3, 2, 1]$, a frozen shuffle $[5, 2, 1, 8, 9, 3, 3, 1]$, and the NLL tie-break $[1, 2, 3, 5, 8, 9, 3, 1]$.

The NLL tie-break minimizes $\sum_a b_a \ell_{N_a - 1}$ among schedules with the optimal robust load, where $b_a = N_{a+1} - N_a$ and ℓ_s is the mean NLL of masked tokens after revealing s reference tokens. We estimate ℓ_s with LLaDA using the first 32 answer tokens of 128 GSM8K training examples sampled uniformly among those with at least 32 answer tokens. We average over prompts and three random coordinate orders per prompt.

Under uniform selection at 32 calls, reversal has the highest accuracy among the simple controls (Figure 5). Robust minus reversed is $+9.64$ points and robust minus equal is $+16.15$ points; the NLL tie-break differs from robust by -0.26 points.

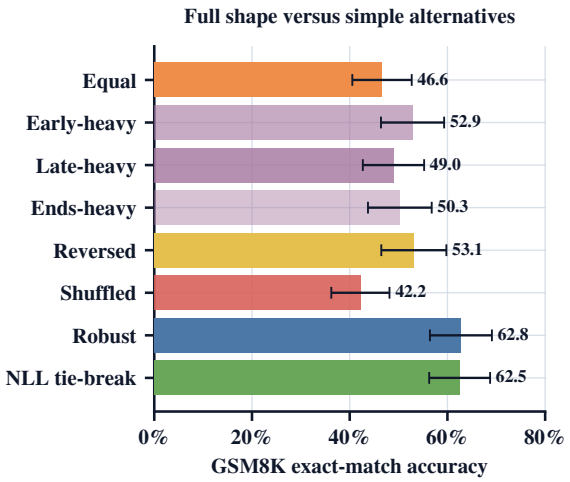

Figure 5: **Full-shape control at NFE 32.** Robust, equal, tapered, reversed, shuffled, and predictor-aware schedules on 128 prompts and three seeds. Error bars show 95% intervals over prompts.

At seven calls on 128 tokens, equal batching uses $[19, 19, 18, 18, 18, 18, 18]$, and robust batching uses $[2, 4, 8, 16, 32, 64, 2]$. Robust minus equal batching is $+16.02$ points under both the frozen DUS-derived ranking and the entropy ranking, and robust minus native entropy-adaptive batch sizes is $+16.41$ points. Native entropy-bounded unmasking (Ben-Hamu et al., 2025) commits an adaptive lowest-entropy prefix. Its threshold was set on the development split to give about seven mean calls; on the test split it averages 7.17 calls (4 to 10), whereas fixed schedules use exactly seven calls per prompt.