COUNT THE GROUPS, NOT THE CONFIGURATIONS: EXACT INFERENCE AND LEARNING AT FIXED TOTALS

Anonymous authors
Paper under double-blind review

ABSTRACT

A fixed total couples every variable in an integer allocation model. For interactions determined by group totals, we turn this constraint into a computational resource: sum local allocation weights into group-count tables, remove a redundant count coordinate, and sample exact configurations from the resulting small distribution. We reuse these tables to score whole batches of changes to group membership through exact partition-function ratios. For two groups, compilation costs $O(Jc_{\max}r)$ for J variables, maximum capacity $c_{\max}$, and total r, versus $O(Jc_{\max}r^2)$ for an explicit total-and-signed-count recursion. A shared conditional allocation law also makes model comparison lossless at the count level. Controlled synthetic experiments with heterogeneous local weights yield 1.48–$2.98\times$ more effective samples per CPU second than the best tested positive auxiliary-field mixtures, and count-compressed membership scoring is $1.75\times$ faster than its expanded implementation. Learning studies separate this computational benefit from the choice of statistical objective. Exact residual calculations further reveal how an omitted interaction can amplify rare configurations, damaging importance sampling before substantially changing the bulk distribution. The result is a reusable count representation for sampling, comparison, and discrete structure refinement.

1 INTRODUCTION

Suppose a system must allocate exactly r units across J locations. Marginal forecasts do not specify how to distribute the *whole allocation* while respecting the total and interactions between locations. An interaction may favor allocations concentrated in either of two groups, while individual locations retain different propensities within each group. Even a binary allocation, with at most one unit per location, has more than 10^{34} feasible configurations when $J = 120$ and $r = 54$. Directly summing over them is impractical.

There is, however, a much smaller question hidden inside this large one: *how many units go to each group?* If the interaction depends only on group totals, then all configurations sharing those totals receive the same interaction weight. Their different local propensities can be summed exactly by polynomial coefficients. Once the group totals have been drawn, an ordinary conditional allocation completes the sample. Figure 1 shows the resulting computation.

The fixed total is decisive. Let x_j be the allocation at location j, and $g_j \in \{-1, +1\}$ its group label. If K units go to the positive group, then $g^\top x = 2K - r$: one count determines the interaction. Organizing the computation by groups replaces a recursion over allocated total and signed total with one-dimensional coefficient tables. For maximum capacity $c_{\max}$, the work falls from $O(Jc_{\max}r^2)$ for the explicit two-coordinate recursion to $O(Jc_{\max}r)$. The feasible allocations and their probabilities stay unchanged.

We call this precomputation a *group-count compiler*: it stores weighted counts for reuse, rather than enumerating complete allocations. The resulting normalizing constant, or partition function, supports likelihood fitting as well as sampling. The same tables can score changes to an unknown grouping or represent a dominant interaction while importance weights account for the remainder.

Our central contribution is this reusable computational organization: exact group-count inference, lossless count-level comparison under a shared allocation law, and a count-compressed calculation of all eligible membership-swap ratios. We study when to compile an interaction fully, when to

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.

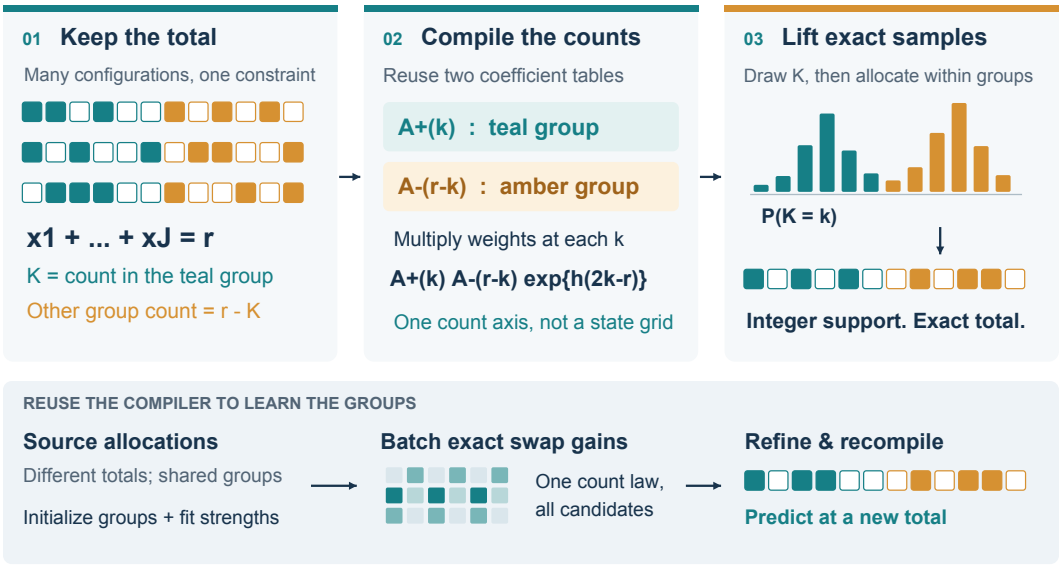


Figure 1: **Use the total to remove a dimension; reuse the counts to learn.** Top: the compiler stores weighted group counts $A_+(k)$ and $A_-(r-k)$, samples their interaction-weighted combination, and lifts the count to an exact configuration. Heterogeneous local weights remain in the coefficient tables. Bottom: source allocations initialize a grouping; the current count law scores all eligible membership swaps, refining the model for prediction at a new total.

correct a simpler representation by importance sampling, and how to reuse exact inference during learning. Controlled experiments measure sampling efficiency, rare-tail behavior, and prediction at a new total. The story is therefore not simply a faster sampler: one representation supports the full *compile–sample–refine* workflow.

2 Related Work

Conditional counts and exact allocation. Weighted binary sampling conditional on a sum is the conditional Bernoulli distribution. Its coefficient recursions and exact sampling algorithms are classical (Chen & Liu, 1997); efficient swap-chain alternatives have also been analyzed (Heng et al., 2020). Sequential importance sampling extends support-aware generation to constrained contingency tables (Chen et al., 2006). We use these computations inside groups. The additional layer is an interacting distribution *over* group counts under the same total.

Cardinality, exchangeability, and lifted inference. Recursive cardinality models provide exact marginalization and sampling through auxiliary count trees and convolution (Tarlow et al., 2012). Exchangeable variable models exploit count representations for tractable learning and inference (Niepert & Domingos, 2014). Lifted inference similarly reasons about classes of individuals (Poole, 2003), including through first-order knowledge compilation (Van den Broeck et al., 2011). Here shared interaction signatures define the classes, while coefficient tables retain unequal site weights. Our learning query is to batch exact normalizer ratios for many *changes to group membership*, reusing the current count law.

Elimination and compiled inference. Variable elimination (Dechter, 1999) and sum-product message passing (Kschischang et al., 2001) organize exact inference around intermediate summaries. Arithmetic-circuit inference (Darwiche, 2003) and weighted model counting (Chavira & Darwiche, 2008) emphasize compiling a model for reusable queries. Our coefficient recursion is a specialized sum-product computation: it eliminates site assignments while retaining group counts. The fixed total then removes one count coordinate. This specialization yields an explicit cost and an all-candidate membership-swap interface.

Group structure and low-rank energies. Multi-species mean-field models use group magnetizations to represent interactions, and their inverse problems include multimodal parameter reconstruction (Fedele & Vernia, 2017). Low-rank Ising sampling can instead introduce auxiliary Gaussian fields and combine variational methods with Markov chains (Koehler et al., 2022). Here the variables have a hard cardinality constraint and heterogeneous local weights. We compare exact count compilation with positive auxiliary-field mixtures on this same support; group-count enumeration and auxiliary fields are the two computational representations under study.

Learning energy models. Likelihood gradients compare observed and model moments (Wainwright & Jordan, 2008). Alternatives include contrastive divergence (Hinton, 2002), score matching for continuous densities (Hyvärinen, 2005), noise-contrastive estimation (Gutmann & Hyvärinen, 2010), and importance estimates in autoregressive energy models (Nash & Durkan, 2019). Pseudolikelihood (PL) (Besag, 1975) multiplies local conditional probabilities instead of the full joint density and belongs to composite likelihood (Varin et al., 2011). Analyses cover spin systems (Chatterjee, 2007) and hard-constrained Ising models (Chauhan & Panageas, 2026). We compare joint-likelihood and pair-conditional objectives within the same discrete structure class.

Importance efficiency. Importance-sampling cost depends on proposal–target discrepancy (Agapiou et al., 2017; Chatterjee & Diaconis, 2018), while effective sample size (ESS) admits several definitions (Martino et al., 2017). We compute a population second-moment fraction and contrast it with Kullback–Leibler (KL) divergence. Exact count calculations expose how rare configurations affect these diagnostics differently.

3 Fixed-total models and their compressed representation

Let $c = (c_1, \dots, c_J)$ be nonnegative integer capacities and $r \in \{0, \dots, \sum_j c_j\}$ the total. An allocation x belongs to the feasible support

$$\mathcal{X}(c, r) = \left\{ x \in \mathbb{Z}^J_{\geq 0} : x_j \leq c_j, \quad \sum_{j=1}^{J} x_j = r \right\}. \tag{1}$$

Write $b_j(a) > 0$ for a local weight on allowed values. Examples are $b_j(a) = \binom{c_j}{a} \exp(u_j a)$ and binary weights $b_j(a) = \exp(u_j a)$, where u_j is a location-specific log propensity, also called a *unary field*. Consider the energy model

$$p(x \mid c, r) = \frac{\mathbb{I}\{x \in \mathcal{X}(c, r)\}}{Z(c, r)} \left[\prod_{j=1}^{J} b_j(x_j) \right] \exp\{H(G^\top x) + R(x)\}, \tag{2}$$

Here $\mathbb{I}$ is an indicator, $G \in \mathbb{R}^{J \times d}$ assigns each location a signature in a small finite set, and $G^\top x$ contains d weighted group totals. The finite-valued functions H and R are the compiled interaction and remaining energy; $Z(c, r)$ is the sum of unnormalized weights over the support. For two groups, $G = g \in \{-1, +1\}^J$ and $H(s) = \beta s^2 / J$ with strength β. We call observed training totals *source* totals, and a total at which the fitted law is evaluated a *target* total.

3.1 Compile local heterogeneity; enumerate group totals

Partition locations into classes $I_1, \dots, I_m$ with equal rows of G. Let $\gamma_a \in \mathbb{R}^d$ be the transpose of their common row and $k_a = \sum_{j \in I_a} x_j$. Using $[z^k]$ to extract a polynomial's coefficient of z^k, define

$$A_a(k) = [z^k] \prod_{j \in I_a} \left(\sum_{v=0}^{c_j} b_j(v) z^v \right), \qquad \mathcal{K}_r = \{k \in \mathbb{Z}^m_{\geq 0} : \textstyle\sum_a k_a = r,\ k_a \leq \sum_{j \in I_a} c_j\}. \tag{3}$$

The count vector k is a *macrostate*; an allocation x is a configuration. $A_a(k)$ sums all local weights in class a at count k, retaining unequal unary fields.

Proposition 1 (Exact group-count compilation)**.** *For $R = 0$, the normalizer and group-count law are*

$$Z(c, r) = \sum_{k \in \mathcal{K}_r} \left[\prod_{a=1}^{m} A_a(k_a) \right] \exp\left\{ H\left(\sum_a \gamma_a k_a \right) \right\}, \tag{4}$$

$$p(k \mid c, r) = \frac{1}{Z(c, r)} \left[\prod_a A_a(k_a) \right] \exp\left\{ H\left(\sum_a \gamma_a k_a \right) \right\}. \tag{5}$$

Conditional on k, the groups are independent, with

$$p(x_{I_a} \mid k_a) = \frac{\mathbb{I}\{\sum_{j \in I_a} x_j = k_a\}}{A_a(k_a)} \prod_{j \in I_a} b_j(x_j).$$

Grouping the sum defining Z by k proves the proposition. Sampling k and then its conditional allocation gives an exact configuration, a step we call *lifting*. Algorithm 1 implements this decomposition, building on macrostate representations (Fedele & Vernia, 2017) while retaining site-specific weights.

Corollary 1 (Lossless count-level comparison)**.** *Let p and q share the support, classes, and local weights, with $R = 0$, but have different H. If p_K, q_K are their count laws, then*

$$\mathrm{KL}(p \| q) = \mathrm{KL}(p_K \| q_K), \qquad \mathbb{E}_q[(p/q)^2] = \mathbb{E}_{q_K}[(p_K/q_K)^2]. \tag{6}$$

The conditional allocation law cancels in the density ratio. Thus compression preserves both prediction discrepancy and importance-weight variability, not only the normalizer. Appendix A gives the proof and the extension to a common refinement of two groupings.

3.2 The total removes one state coordinate

For $g \in \{-1, +1\}^J$, set $I_+ = \{j : g_j = +1\}$, $I_- = I_+^c$, and $K = \sum_{j \in I_+} x_j$. The other count is $r - K$, so

$$g^\top x = 2K - r, \qquad Z(c, r) = \sum_{k=0}^{r} A_+(k) A_-(r - k) \exp\{H(2k - r)\}. \tag{7}$$

Out-of-range coefficients are zero. A two-dimensional recursion that tracks partial total and signed sum can therefore be replaced by two one-dimensional coefficient tables and a sum of at most $r + 1$ terms.

For a class ordered as $(j_1, \ldots, j_n)$, let $B_t(k)$ sum the weights for locations $j_t, \ldots, j_n$ with remaining count k. Then

$$B_t(k) = \sum_{v=0}^{c_{j_t}} b_{j_t}(v) B_{t+1}(k - v), \qquad B_{n+1}(k) = \mathbb{I}\{k = 0\}, \tag{8}$$

$$\Pr(X_{j_t} = v \mid k) = \frac{b_{j_t}(v) B_{t+1}(k - v)}{B_t(k)}. \tag{9}$$

Each draw reduces the remaining count by v. Feasibility is built into the recursion rather than repaired after sampling. In practice the tables are stored in log space, and impossible entries are represented by $-\infty$.

Corollary 2 (Two-group cost)**.** *For $r, c_{\max} = \max_j c_j \geq 1$ and constant-cost H, suffix compilation takes $O(J c_{\max} r)$ operations and $O(Jr)$ storage. Drawing S configurations takes $O(r + S J c_{\max})$ additional work. The corresponding rectangular total-and-signed-count recursion uses $O(J c_{\max} r^2)$ work and $O(Jr^2)$ storage.*

For binary variables, allocation takes $O(J)$ per draw. At $r = 0$ the allocation is deterministically zero. Tables are reusable across draws and changes to H with fixed local weights and groups.

Algorithm 1 Exact sampling by group counts

Require: Capacities c, weights b_j, classes I_a, signatures γ_a, total r, energy H
Ensure: Exact sample x, log probability $\log p(x)$, normalizer Z $\quad (R = 0)$

1: Build suffix tables $B_{a,t}$ by Eq. (8); set $A_a(k) = B_{a,1}(k)$.
2: For each $k \in \mathcal{K}_r$, compute $w(k) = \prod_a A_a(k_a) \exp\{H(\sum_a \gamma_a k_a)\}$.
3: Set $Z = \sum_{k \in \mathcal{K}_r} w(k)$ and draw K with probability $w(K)/Z$.
4: For each class a, draw x_{I_a} by Eq. (9) with remaining count K_a.
5: **return** x, $\sum_j \log b_j(x_j) + H(G^\top x) - \log Z$, and Z.

4 Compile the dominant interaction, correct the remainder

Let q be the compiled law with $R = 0$. The full model in Eq. (2) satisfies

$$p(x) = q(x) \frac{\exp R(x)}{\mathbb{E}_q \exp R(X)}, \qquad Z_p = Z_q \mathbb{E}_q \exp R(X). \tag{10}$$

Thus exact draws from q give an unbiased estimator of the normalizer ratio. Expectations under p can be estimated with normalized weights; these self-normalized estimates have finite-sample bias. The proposal remains exactly feasible even when the complete interaction is not compiled. This is ordinary importance correction (Agapiou et al., 2017), with the group compiler providing the tractable reference law.

How much of the interaction should be compiled? Define the population effective-sample fraction

$$\mathrm{ESS}_\infty(p, q) = \frac{1}{\mathbb{E}_q[(p(X)/q(X))^2]} = \frac{1}{1 + \chi^2(p\|q)} = \frac{(\mathbb{E}_q e^R)^2}{\mathbb{E}_q e^{2R}}. \tag{11}$$

Here $\chi^2(p\|q) = \sum_x p(x)^2/q(x) - 1$ and $\mathrm{KL}(p\|q) = \sum_x p(x) \log[p(x)/q(x)]$; logarithms are natural. ESS_∞ is a population weight diagnostic, not a sample-estimated ESS or a guarantee for every test function. Another compiled factor costs more but can reduce residual weight fluctuations.

Proposition 2 (A residual-range guarantee)**.** *If $a \le R(x) \le b$ on the common finite support, then*

$$\mathrm{ESS}_\infty(p, q) \ge \frac{4e^{b-a}}{(1 + e^{b-a})^2} = \mathrm{sech}^2\left(\frac{b-a}{2}\right). \tag{12}$$

Here $\mathrm{sech}(z) = 1/\cosh(z)$. The bound uses the *residual energy range*: even one omitted factor can heavily upweight a rare configuration. Its bounded-weight proof is in Appendix A.

For two signed factors g, v, four classes $(g_j, v_j) \in \{-1, +1\}^2$ suffice for exact full compilation. In the binary quadratic family used below,

$$p_r(x) \propto \mathbb{I}\{\textstyle\sum_j x_j = r\} \exp\left\{u^\top x + \frac{\beta_g}{J}(g^\top x)^2 + \frac{\beta_v}{J}(v^\top x)^2\right\}. \tag{13}$$

The coefficients β_g, β_v control interaction strengths. One-factor compilation retains g and leaves $R(x) = \beta_v(v^\top x)^2/J$ to importance weights; four-class compilation retains both. For a small residual, the former trades exact representation of the full law for cheaper samples with corrective weights.

5 Use the compiler to learn group structure

The same normalizer supports conditional likelihood on source data $\mathcal{D} = \{(x^{(n)}, r_n, u_n)\}_{n=1}^N$, where u_n is the known unary vector for observation n. Totals and unaries may vary; the grouping is shared. For a fixed grouping, source likelihood fits interaction strengths by matching observed and model-expected group statistics.

Exact membership swaps. Write p_v and Z_v for the law in Eq. (13) and its normalizer with $g, u, r, \beta_g, \beta_v$ fixed. Choose i, j with $g_i = g_j$, $v_i = +1$, $v_j = -1$, and swap these two signs to obtain v'. This preserves class sizes. With $s = v^\top x$ and $d = x_j - x_i$,

$$s' = s + 2d, \qquad \Delta E_{ij}(x) = \frac{4\beta_v}{J}(sd + d^2), \qquad \frac{Z_{v'}}{Z_v} = \mathbb{E}_{p_v} \exp\{\Delta E_{ij}(X)\}. \tag{14}$$

For N_r source observations sharing a total r and unary vector, the exact log-likelihood gain is

$$\Delta\ell_{ij} = \sum_{n=1}^{N_r} \Delta E_{ij}(x^{(n)}) - N_r \log \mathbb{E}_{p_v} \exp\{\Delta E_{ij}(X)\}. \tag{15}$$

Gains add across source totals and unary vectors. Crucially, all candidate swaps are expectations under the *current* compiled model. They do not each require a new partition-function computation.

Given a macrostate, the two sites belong to distinct classes and are independent. Their inclusion probabilities follow from coefficient ratios. Since $d \in \{-1, 0, 1\}$, only three cases enter the expectation. We initialize groups by source principal component analysis (PCA), fit strengths, accept positive-gain swaps, and recompile after each accepted update. Alternating swaps and strength fitting stops when no allowed swap improves the fixed-strength objective, or at the iteration cap.

Compress the batch, too. There is a second redundancy: a site's inclusion probability depends only on its own class count. For eligible classes A, B containing sites i, j, define $U_{a,i} = \Pr(X_i = 1 \mid k_A = a)$ and $V_{b,j} = \Pr(X_j = 1 \mid k_B = b)$. Let $s_v(k)$ be the second signed total determined by k. With $\lambda = \beta_v/J$, form

$$D^{\pm}_{ab} = \sum_{k:\, k_A=a,\, k_B=b} p_v(k)\big(e^{4\lambda(1\pm s_v(k))} - 1\big). \tag{16}$$

Conditional independence then gives *every* candidate normalizer ratio in one matrix:

$$\left[\frac{Z_{v^{(ij)}}}{Z_v}\right]_{ij} = \mathbf{1} + U^\top D^-(\mathbf{1} - V) + (\mathbf{1} - U)^\top D^+ V. \tag{17}$$

Here $v^{(ij)}$ is the swapped labeling and each $\mathbf{1}$ is an all-ones matrix of the required shape. $D^{\pm}$ are $(r+1) \times (r+1)$, while U, V have one column per site in their class. For $M = |\mathcal{K}_r|$, this reduces the ratio stage from expanded $O(MJ^2)$ cross-products to $O(M + Jr^2 + J^2 r)$ work with inclusion tables available. The macrostate sum is performed once per eligible class pair (proof in Appendix B).

Separating structural recovery from prediction. Two groupings can differ at a few sites yet give similar laws. We evaluate $\mathrm{KL}(p_{\mathrm{true}} \| p_{\mathrm{fitted}})$ at the target total. For shared unaries, refining classes by both groupings gives a common conditional allocation law, so Corollary 1 applies. Known-energy sampling instead supplies the grouping and strengths, separating representation cost from estimation error.

6 Experiments

Controlled synthetic models supply exact reference probabilities. A *structure* (also called a graph instance below) is a randomly drawn group labeling; a *stream* is an independent source sample conditional on that labeling. Balanced labels have equal numbers of signs; orthogonal factors satisfy $g^\top v = 0$.

6.1 Protocol and comparators

Known-energy inference. We test binary $J \in \{80, 120, 160\}$ systems with nonuniform unary fields. The symmetric cases have balanced groups and $\beta = 2.5$; a fresh asymmetric $J = 120$ case uses $\beta = 2.8$. Each timing includes compilation, 4,096 draws, and evaluation of proposal densities. Five CPU timings are recorded per configuration. We report $\mathrm{ESS}_\infty \times 4096/T$, with T the measured CPU cost.

Positive field mixtures. For $\beta \geq 0$, $e^{\beta s^2/J} = \mathbb{E}_{\xi\sim\mathcal{N}(0,1)} e^{\sqrt{2\beta/J}\,\xi s}$ motivates mixtures of fixed-total laws with additive log weights. We compare Gaussian quadrature with $Q \in \{2, 4, 8, 16, 32, 64, 128\}$ nodes and optimized positive-weight mixtures with $Q \in \{1, 2, 4, 8, 16\}$ components. The latter use the true target law to minimize $\chi^2(p\|q)$; fitting cost is excluded, while sampling and density evaluation are charged. Appendix D also describes cached mean-field and moment-based proposals for bounded counts.

Table 1: **Known-energy inference.** The best positive mixture is selected from all tested Gaussian and target-informed sparse mixtures. Speed ratios compare population effective samples per CPU, including compilation and density evaluation. The compiler's ESS is one in these exactly represented cases.

Setting	J	Best mixture Q	Mixture ESS	Compiler / mixture
Balanced	80	4	0.6531	$2.568\times$
Balanced	160	8	0.9868	$2.983\times$
Asymmetric	120	4	0.9962	$1.477\times$

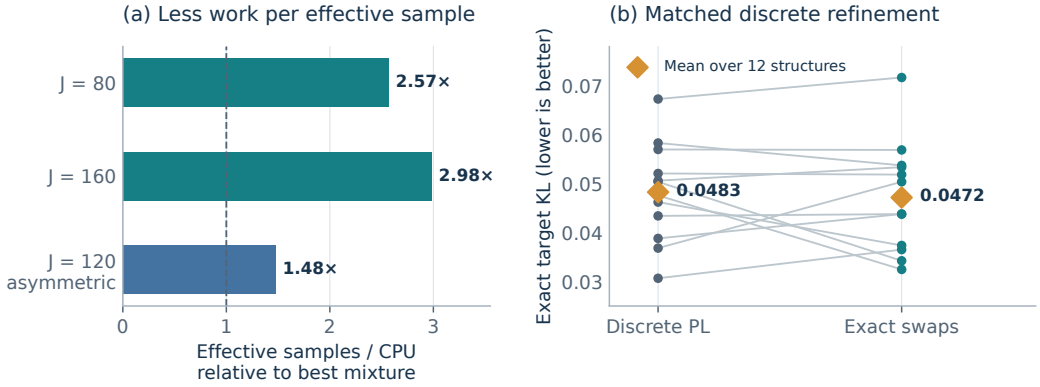


Figure 2: **Computational gains and a matched learning comparison.** Left: exact group compilation versus the best tested positive field mixture. Right: target KL on twelve fresh structures, each averaged over two source streams. Lines pair the same structure across learners. Both use known local fields; discrete PL additionally receives true interaction strengths during structure search.

Source-learned groups. Four balanced $J = 120$ group assignments generate three streams at each of two sample sizes: 30 or 100 observations per source total $42, 60, 78$, giving 24 fits. Unaries are known and vary by total. Within-total-centered PCA estimates groups; source likelihood fits β. Evaluation at total 54 is interpolation across totals within this model family, without target observations for fitting.

Correctness checks. Feasible sets of 252 or 924 allocations are enumerated to check normalizers, probability mass, proposal density, and importance identities. A separate coefficient calculation checks macrostate results. Compact and original recursions agree in log density to 1.4×10^{-14}. All timings use single-threaded CPU implementations (Appendix D).

6.2 Compression improves exact sampling efficiency

Table 1 isolates the benefit of the representation. The compiler is exact and faster per effective sample even when the field mixture is given the target law during optimization. At $J = 160$, the winning mixture already has ESS 0.9868: the remaining advantage is mostly the cost of sampling and evaluating several additive components. In the asymmetric case, joint optimization of field locations and weights gives the best mixture ESS 0.9962, and the compiler retains a $1.477\times$ advantage.

On bounded-count models with learned interactions, the compressed and original signed-count recursions define the same proposal. The compressed implementation is faster in all 24 planted cases, with mean speedups $1.75\times$, $2.61\times$, and $3.82\times$ at capacities 2, 3, and 4. Against the best cached conventional proposal, mean effective-sample/CPU ratios are 4.89, 2.56, and 1.78. Random dense controls range from 0.83 to 1.10, consistent with gains concentrated in group-structured interactions (complete table in Appendix D).

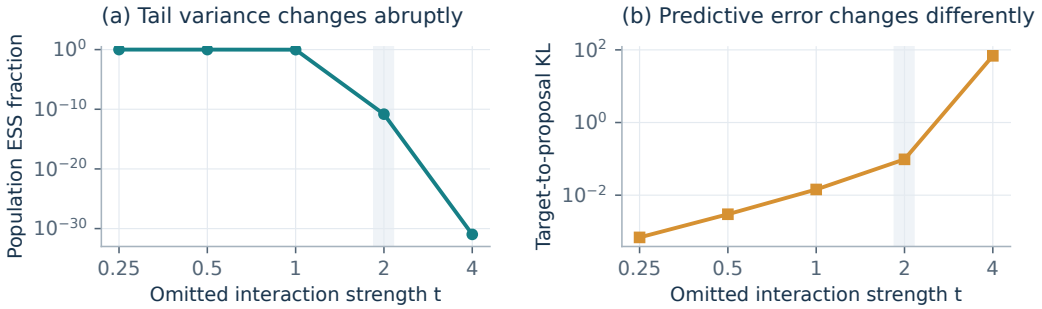


Figure 3: **A rare tail can dominate importance variance before it dominates prediction.** Exact evaluations for a dominant-factor proposal as the omitted interaction grows. The shaded $t = 2$ case has tiny population ESS but moderate KL; at $t = 4$ both diagnostics change substantially. Curves join the five evaluated strengths, not a continuous sweep.

6.3 Learning the groups preserves the computation advantage

All 24 pure-block fits recover the planted partition up to sign. At 30 observations per source total, mean exact target KL is 0.001792 and the minimum population ESS is 0.9734; at 100 observations, they are 0.002403 and 0.9842. Relative to the best target-informed true-group mixture, the median efficiency ratios are $1.863\times$ and $1.866\times$, with all 24 ratios between 1.751 and 1.970. Thus in this family, the benefit of count compilation survives source estimation by ordinary PCA and exact coefficient fitting.

6.4 Partial compilation exposes two different error scales

We add a second orthogonal signed interaction to a dominant $\beta_g = 2.8$ interaction at $J = 120$, $r = 54$, and vary its strength $t = \beta_v$. For $t = 0.25, 0.5, 1$, compiling only the dominant factor yields population ESS $0.9986, 0.9935, 0.9639$. Effective samples per CPU are 1.65–$1.70\times$ those of the best tested target-informed two-field mixture and about $2.3\times$ those of full two-factor compilation. This is the regime in which an inexpensive exact reference plus residual weights is attractive.

The exact macrostate calculation also distinguishes *tail reliability* from *distributional approximation* (Figure 3). At $t = 2$, the event $|v^\top X| \geq 30$ has probability 3.90×10^{-5} under the target but 6.77×10^{-12} under the proposal. Population ESS falls to 1.50×10^{-11}, even though target-to-proposal KL is only 0.097 and total variation, $\frac{1}{2}\sum_x |p(x) - q(x)|$, is 0.122. At $t = 4$, that event carries nearly all target mass, KL is about 68.6, and total variation approaches one. Compiling the second factor resolves both cases. The distinction matters: the $t = 2$ phenomenon is a severe second-moment penalty, whereas $t = 4$ also moves the bulk of the predictive law.

6.5 From spectral groups to discrete likelihood refinement

We use the compiled normalizers to refine the $t = 2$ second factor at $J = 120$. An initial study on six fresh balanced orthogonal factor pairs, with three source streams per pair, compares exact swaps with spectral initialization and continuous-fit-then-project exchange pseudolikelihood. All methods observe 100 tables at each source total $42, 60, 78$ and predict at total 54 with known heterogeneous unary fields. Exact swaps reduce mean target KL from 0.041220 (spectral) and 0.039390 (projected PL) to 0.032677, improving all six graph averages. Appendix C preserves this comparison and its information conditions.

To isolate the learning objective from continuous projection, we froze a comparison on twelve new factor pairs with two streams each. Both learners search the same balanced discrete second-factor class. Exact joint likelihood starts from source PCA and alternates swaps with source coefficient fitting. Discrete PL scores pairwise allocation exchanges conditional on all other sites; its exchanges concern observations, not group labels (Appendix C). It receives the same start *and* the best projected continuous start, chooses by its source objective, and uses true strengths during structure search. Both

Table 2: **Matched discrete learners at a new total.** Exact target KL on twelve new structures, two source streams each. The protocol was frozen before these instances were generated. Graphs, rather than streams, are the replication units.

Source fitting procedure	Mean target KL ↓
Discrete pair-exchange PL (oracle strengths)	0.048261
PCA + exact joint-likelihood swaps	0.047168

refit coefficients by exact source likelihood for prediction. Thus PL has additional information and initialization within the same structure class.

Table 2 and Figure 2 show the resulting predictive comparison. The mean PL-minus-joint KL difference is 0.001093, with a descriptive graph-paired 95% t-interval of $[-0.004472, 0.006659]$. Joint likelihood wins on 6/12 graph means and 11/24 streams, with four stream ties. Both searches terminate without reaching their iteration caps. The observed means are close in this structure-class-matched comparison; the compiler supplies exact joint-likelihood evaluation and samples from either fitted model.

The count-compressed swap calculation in Eq. (17) preserves all candidate gains to 5.4×10^{-13} across six structures and three source totals. Ten interleaved timing repetitions per condition give a median $1.75\times$ speedup over the expanded macrostate implementation (range 1.68–$2.09\times$). This measures the complete swap-scoring call, including macrostate construction, rather than end-to-end training.

7 Conclusion and discussion

Fixed totals do more than define a feasible set: they reveal redundant coordinates in the inference problem. For signed group energies, compiling group counts removes one state dimension while preserving exact probabilities and samples. The same computation supports source likelihood, residual importance correction, and discrete structure comparisons. The experiments establish gains in sampling and swap-scoring efficiency and show how the representation can be reused after learning. Exact inference is useful here not only as an endpoint, but as a reusable component of learning.

Scope and next steps. The evidence is synthetic and class-specific, with supplied unary fields and group-class constraints. Group counting and compiled inference have substantial precedents; the contribution is the fixed-total computational organization and its batched membership-query interface. The twelve-graph comparison does not establish predictive superiority over strong discrete PL: the earlier 17% gain applies to the projected continuous control. The sampler comparison covers tested positive mixtures, not all MCMC or circuit methods, and local search need not find the global optimum. Future work can learn richer local weights, extend the interaction class, and choose which factors to compile using a criterion matched to the eventual estimation task.

Reproducibility statement

The appendix provides coefficient recursions, proofs, swap formulas, and protocols. Figures and headline values are reconstructed from saved synthetic receipts. The six- and twelve-structure studies each used protocols fixed before their respective instances were generated; their distinct comparators and roles are recorded separately.

AI use statement

AI assistants were used extensively in the research workflow, including method exploration, mathematical derivations, code generation, numerical checking, literature search, manuscript drafting, and figure production. The authors have independently verified all content, item by item, and take responsibility for the final claims and submission.

REFERENCES

Sergios Agapiou, Omiros Papaspiliopoulos, Daniel Sanz-Alonso, and Andrew M. Stuart. Importance sampling: Intrinsic dimension and computational cost. *Statistical Science*, 32(3):405–431, 2017. doi: 10.1214/17-STS611.

Julian Besag. Statistical analysis of non-lattice data. *The Statistician*, 24(3):179–195, 1975. doi: 10.2307/2987782.

Sourav Chatterjee. Estimation in spin glasses: A first step. *The Annals of Statistics*, 35(5):1931–1946, 2007. doi: 10.1214/009053607000000109.

Sourav Chatterjee and Persi Diaconis. The sample size required in importance sampling. *The Annals of Applied Probability*, 28(2):1099–1135, 2018. doi: 10.1214/17-AAP1326.

Rohan Chauhan and Ioannis Panageas. Learning the inverse temperature of Ising models under hard constraints using one sample. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2509.20993.

Mark Chavira and Adnan Darwiche. On probabilistic inference by weighted model counting. *Artificial Intelligence*, 172(6–7):772–799, 2008. doi: 10.1016/j.artint.2007.11.002.

Sean X. Chen and Jun S. Liu. Statistical applications of the Poisson-Binomial and conditional Bernoulli distributions. *Statistica Sinica*, 7(4):875–892, 1997. URL https://www3.stat.sinica.edu.tw/statistica/j7n4/j7n44/j7n44.htm.

Yuguo Chen, Ian H. Dinwoodie, and Seth Sullivant. Sequential importance sampling for multiway tables. *The Annals of Statistics*, 34(1):523–545, 2006. URL https://arxiv.org/abs/math/0605615.

Adnan Darwiche. A differential approach to inference in Bayesian networks. *Journal of the ACM*, 50 (3):280–305, 2003. doi: 10.1145/765568.765570.

Rina Dechter. Bucket elimination: A unifying framework for reasoning. *Artificial Intelligence*, 113 (1–2):41–85, 1999. doi: 10.1016/S0004-3702(99)00059-4.

Micaela Fedele and Cecilia Vernia. Inverse problem for multispecies ferromagneticlike mean-field models in phase space with many states. *Physical Review E*, 96:042135, 2017. doi: 10.1103/PhysRevE.96.042135.

Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pp. 297–304, 2010. URL https://proceedings.mlr.press/v9/gutmann10a.html.

Jeremy Heng, Pierre E. Jacob, and Nianqiao Ju. A simple Markov chain for independent Bernoulli variables conditioned on their sum. *arXiv preprint arXiv:2012.03103*, 2020. URL https://arxiv.org/abs/2012.03103.

Geoffrey E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800, 2002. doi: 10.1162/089976602760128018.

Aapo Hyvärinen. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6:695–709, 2005. URL https://jmlr.org/papers/v6/hyvarinen05a.html.

Frederic Koehler, Holden Lee, and Andrej Risteski. Sampling approximately low-rank Ising models: MCMC meets variational methods. In *Proceedings of the Conference on Learning Theory*, volume 178 of *Proceedings of Machine Learning Research*, pp. 4945–4988, 2022. URL https://proceedings.mlr.press/v178/koehler22a.html.

Frank R. Kschischang, Brendan J. Frey, and Hans-Andrea Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47(2):498–519, 2001. doi: 10.1109/18.910572.

Luca Martino, Víctor Elvira, and Francisco Louzada. Effective sample size for importance sampling based on discrepancy measures. *Signal Processing*, 131:386–401, 2017. doi: 10.1016/j.sigpro.2016.08.025.

Charlie Nash and Conor Durkan. Autoregressive energy machines. In *Proceedings of the International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 1735–1744, 2019. URL https://proceedings.mlr.press/v97/durkan19a.html.

Mathias Niepert and Pedro Domingos. Exchangeable variable models. In *Proceedings of the International Conference on Machine Learning*, volume 32(2) of *Proceedings of Machine Learning Research*, pp. 271–279, 2014. URL https://proceedings.mlr.press/v32/niepert14.html.

David Poole. First-order probabilistic inference. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 985–991, 2003. URL https://www.cs.ubc.ca/labs/lci/papers/docs2003/fopi.html.

Daniel Tarlow, Kevin Swersky, Richard S. Zemel, Ryan Prescott Adams, and Brendan J. Frey. Fast exact inference for recursive cardinality models. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pp. 825–834, 2012. URL https://arxiv.org/abs/1210.4899.

Guy Van den Broeck, Nima Taghipour, Wannes Meert, Jesse Davis, and Luc De Raedt. Lifted probabilistic inference by first-order knowledge compilation. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 2178–2185, 2011. URL https://starai.cs.ucla.edu/papers/VdBIJCAI11.pdf.

Cristiano Varin, Nancy Reid, and David Firth. An overview of composite likelihood methods. *Statistica Sinica*, 21(1):5–42, 2011. URL https://www3.stat.sinica.edu.tw/statistica/oldpdf/A21n11.pdf.

Martin J. Wainwright and Michael I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1–2):1–305, 2008. doi: 10.1561/2200000001.

A Exact computation and proofs

A.1 Coefficient compilation and exact lifting

For each class I_a, expanding the product in Eq. (3) gives

$$A_a(k) = \sum_{x_{I_a} : \sum_{j \in I_a} x_j = k} \prod_{j \in I_a} b_j(x_j).$$

For any configuration with class counts k, the interaction is $H(\sum_a \gamma_a k_a)$. Partitioning the global sum over configurations by these counts proves Proposition 1. Dividing each class weight by $A_a(k_a)$ proves conditional independence of the groups.

For Eq. (9), the numerator is the weight of choosing v times the summed weight of every compatible continuation. The denominator sums the numerator over allowed v. Multiplying these successive conditional probabilities telescopes the suffix coefficients, leaving $\prod_j b_j(x_j)/A_a(k_a)$. Multiplication by the sampled macrostate probability then returns the original normalized model. This proves exactness of both the sampling distribution and its scored density.

There are at most $J(r+1)$ suffix states, and each sums at most $c_{\max} + 1$ terms. Storing suffix rather than only prefix coefficients permits subsequent sampling without rebuilding the tables. For two classes the macrostate sum has at most $r + 1$ terms. For m signatures, at most $(r+1)^{m-1}$ count combinations satisfy one total. Thus complexity depends on the number of distinct signatures as well as the number of sites. Out-of-range coefficient entries are zero, and their conditional inclusion probabilities can be assigned zero because such macrostates have zero mass.

A.2 Why count-level comparison is lossless

Proposition 1 factors the laws as $p(x) = p_K(k)L(x \mid k)$ and $q(x) = q_K(k)L(x \mid k)$, where L is the shared within-class allocation law. Positivity on the common support implies $p(x)/q(x) = p_K(k)/q_K(k)$. Summing this log ratio under p proves the KL identity in Corollary 1; summing its square under q proves the second-moment identity. More generally, every f-divergence $D_f(p\|q) = \sum_x q(x) f(p(x)/q(x))$, for convex f with $f(1) = 0$, equals $D_f(p_K\|q_K)$ whenever the same conditional law cancels. In particular, total variation $\frac{1}{2}\sum_x |p(x) - q(x)|$ is also preserved.

This conditional factorization permits exact distributional comparisons on the compiled state space. If the local weights differ, cancellation need not hold. In that case the KL chain rule reads

$$\mathrm{KL}(p\|q) = \mathrm{KL}(p_K\|q_K) + \sum_k p_K(k)\,\mathrm{KL}\left(p(\cdot \mid k)\|q(\cdot \mid k)\right).$$

Count-only comparison is exact when the conditional term vanishes; otherwise that term must be included. For different groupings with shared local weights, partition sites by their joint signatures. Both interaction energies are constant on each resulting count state, restoring the common conditional law.

A.3 Bounded residuals

Let $W = e^{R(X)}$ under q, with $m = e^a$, $M = e^b$, and $\mu = \mathbb{E}W$. Since $(W - m)(W - M) \le 0$,

$$\mathbb{E}W^2 \le (m + M)\mu - mM.$$

The function $((m + M)\mu - mM)/\mu^2$ attains its maximum for $\mu = 2mM/(m + M)$ on $[m, M]$. Consequently

$$\frac{\mathbb{E}W^2}{(\mathbb{E}W)^2} \le \frac{(m+M)^2}{4mM},$$

which proves Proposition 2. Equality is attainable when W takes only the endpoints m, M with probabilities $M/(m+M), m/(m+M)$. The bound is therefore sharp given only the residual range, although a particular compiled model may have a much larger ESS fraction.

A.4 Several factors and exact comparison of different groupings

For two signed factors, the four signature classes have counts $k_{++}, k_{+-}, k_{-+}, k_{--}$. The signed totals are

$$s_g = k_{++} + k_{+-} - k_{-+} - k_{--},$$
$$s_v = k_{++} - k_{+-} + k_{-+} - k_{--}.$$

Three counts determine the fourth under the fixed total. Its exact macro weight is

$$\prod_a A_a(k_a) \exp\{\beta_g s_g^2/J + \beta_v s_v^2/J\}.$$

To compare a true grouping and an estimated grouping, refine the classes by their joint signatures. Both energies are constant on each refined macrostate. Summing over this common refinement gives exact KL and second moments without enumerating all micro-configurations. The small direct-state audits check this cross-grouping calculation as well as self-normalization.

B Batched partition ratios for structure search

Let i have second-factor sign $+1$ and j sign -1 within the same first-factor group. At a fixed macrostate k, write $s = s_v(k)$ and $a_i(k) = \Pr(X_i = 1 \mid k)$, $b_j(k) = \Pr(X_j = 1 \mid k)$. For binary variables these inclusion probabilities are

$$a_i(k) = e^{u_i} \frac{[z^{k_a - 1}] \prod_{h \in I_a \setminus \{i\}} (1 + e^{u_h} z)}{A_a(k_a)}.$$

They can be obtained from leave-one-out coefficients. Because i and j belong to different signature classes, they are conditionally independent given k. With $\lambda = \beta_v/J$, Eq. (14) yields

$$R_{ij} := \frac{Z_{v'}}{Z_v} = 1 + \sum_k p_v(k) \big[a_i(k)(1 - b_j(k))(e^{4\lambda(1-s)} - 1) + (1 - a_i(k)) b_j(k)(e^{4\lambda(1+s)} - 1) \big]. \tag{18}$$

The $d = 0$ case contributes the unchanged weight, while $d = -1$ and $d = +1$ give the two displayed terms. For all eligible i, j, terms such as $\sum_k p_v(k) e^{\pm 4\lambda s} a_i(k) b_j(k)$ are weighted matrix products. All candidate ratios therefore reuse one current-model macro distribution and its inclusion-probability matrices. The method still pays for these matrices and for each accepted structure's new coefficients; it eliminates rebuilding a full normalizer for *every proposed* swap, not all dependence on the number of variables.

For several source supports, evaluate Eq. (18) on each support and add the corresponding gains in Eq. (15). With coefficients fixed, a strictly positive exact gain increases the finite-source likelihood. Updating coefficients by maximizing that same source likelihood gives an alternating local search. Because the structure class is finite, strictly increasing structure steps cannot revisit a state at fixed coefficients. Neither this fact nor exact gains imply recovery of the generating structure in finite samples.

B.1 Count-compressed all-candidate calculation

In Eq. (18), $a_i(k) = U_{k_A,i}$ and $b_j(k) = V_{k_B,j}$. Grouping the first summand by (k_A, k_B) yields $U^\top D^-(\mathbf{1} - V)$, while grouping the second yields $(\mathbf{1} - U)^\top D^+ V$. This proves Eq. (17). In implementation, three reusable arrays aggregate $p_v(k)$ and $p_v(k) e^{\pm 4\lambda s_v(k)}$; their linear combinations form $D^\pm$. After these $O(M)$ accumulations, multiplication of count-by-count and count-by-site arrays costs $O(Jr^2 + J^2 r)$ for the two eligible class pairs. These costs concern the ratio stage with inclusion tables available, rather than the cost of learning the whole model.

The implementation comparison holds source samples, initial groupings, and source-fitted coefficients fixed across the six fresh structures and all three source totals. It uses the first source stream of each structure, with one warm-up and ten alternating-order CPU timings per implementation. Both calls include macrostate construction and all candidate scores. The maximum ratio difference is 5.4×10^{-15} and the maximum source-gain difference is 5.4×10^{-13}. All 18 condition-level median runtimes improve. These are paired computational checks on existing instances, not additional predictive replications.

C MATCHED DISCRETE LEARNING AND THE EARLIER COMPARISON

Pair-conditional objective. For a binary allocation x and indices i, j with $x_i \neq x_j$, let $x^{\leftrightarrow ij}$ swap their allocations, keeping the total fixed. Writing θ for the model parameters, $E_\theta(x)$ for the unnormalized log weight, and x_{-ij} for the other sites, the pair conditional is

$$p_\theta(x_i, x_j \mid x_{-ij}, x_i + x_j) = \frac{e^{E_\theta(x)}}{e^{E_\theta(x)} + e^{E_\theta(x^{\leftrightarrow ij})}}.$$

Pairs with sum zero or two are deterministic and add zero log loss. The control minimizes the averaged negative log conditional over observed allocations and eligible pairs. These *allocation exchanges* define the PL objective; they differ from the *label swaps* used to search over group structures. Both learners search the same allowed label-swap class.

The first comparison used six new balanced orthogonal factor pairs and three source streams per pair. Exact joint-likelihood swaps were compared with a continuous exchange-pseudolikelihood fit followed by projection into the discrete structure class. The PL fit used three starts and true strengths during structural fitting. All selected structures subsequently received exact source coefficient fitting. The complete first-stage means are:

Procedure	Mean target KL
Single dominant factor	0.109287
Projected second PCA direction	0.041220
Continuous-fit-then-project PL (oracle strengths)	0.039390
Exact joint-likelihood swaps	0.032677
True factors + source coefficient fit	0.001816

Exact swaps improved all six graph averages against spectral and projected-PL controls, with 14/18 and 12/18 stream wins, respectively. Graph-paired KL reductions were 0.00854 and 0.00671, with descriptive 95% t-intervals $[0.00489, 0.01220]$ and $[0.00018, 0.01324]$. All dominant groupings and 2/18 second groupings were recovered. Direct enumeration of a 924-state feasible set agreed with the cross-model KL computation to 1.8×10^{-15}.

The follow-up protocol was frozen before drawing twelve new factor pairs, using graph seeds 202609261100–202609261111 and two streams per graph. Each stream contained 100 tables at each total 42, 60, and 78; target total 54 was used for exact distributional evaluation, not fitting. All unaries were known. The discrete PL learner scored all legal balanced sign swaps, with a 30-accepted-swap cap, from both the shared PCA start and its best projected continuous start. It selected the lower source PL objective and then refitted strengths by exact source likelihood. Joint likelihood alternated exact source coefficient fitting and swaps for at most three rounds. All 24 fits terminated without either cap. The PL candidate-score audit differed from direct evaluation by at most 1.67×10^{-16}.

The twelve-graph mean PL-minus-joint KL difference was 0.0010934 with paired 95% t-interval $[-0.0044718, 0.0066586]$. Exact likelihood won 6/12 graph means and 11/24 streams; four streams tied to 10^{-12}. The prespecified comparison criterion required lower mean KL, at least 9/12 graph wins, and a positive lower interval endpoint. It was not met. The first-stage gain therefore establishes improvement over the particular projected control, not a general advantage of joint likelihood over discrete PL. This follow-up consumed 557.62 CPU seconds; the stored receipt, frozen plan, and source-code hash accompany the manuscript.

D EXPERIMENT DETAILS AND COMPLETE COMPARISON CONTEXT

D.1 BOUNDED-COUNT COMPILATION STUDY

The state-compression experiment used four planted signed-block graphs (two balanced and two 3:7 partitions), two random dense controls, and two source draws per graph. Each fitted unrestricted model had 54 parameters: nine unary and 45 pair-interaction parameters for ten variables. There were 600 distinct source supports and 48 unseen supports. Source fitting used a fixed 128-particle moment proposal; exact source likelihood provided a same-information quality reference. Capacity-2/3/4 evaluation supports contained approximately 9 thousand, 116 thousand, and 857 thousand states

and were enumerated for population ESS. Each proposal drew four batches of 1,024 states; cached mean-field construction was charged once, rather than once per batch.

Table 3: All families and capacities in the bounded-count study. CPU columns are seconds for the compact compiler, original signed-state dynamic program, and cached mean-field (MF) implementation. Efficiency is compact divided by the best conventional proposal in each case, then averaged.

Family	Capacity	Cases	Compact ESS	CPU: compact / old / MF	Efficiency
Planted	2	8	.924	.0103 / .0181 / .0140	4.89
Planted	3	8	.946	.0105 / .0274 / .0157	2.56
Planted	4	8	.959	.0126 / .0482 / .0187	1.78
Dense	2	4	.489	.0100 / .0173 / .0143	.83
Dense	3	4	.616	.0104 / .0271 / .0153	.98
Dense	4	4	.696	.0121 / .0482 / .0183	1.10

All eight planted source fits recovered the group signs. The compact/old proposal log-density difference was at most 1.4×10^{-14}. The original prewritten investment gate required at least $2\times$ efficiency in 18 of 24 planted cases; 10 of 24 met that threshold. We retain that outcome alongside the continuous cost measurements. The threshold is a research-selection criterion, not a mathematical boundary between useful and useless computation.

D.2 Large binary inference and learned groups

The $J = 80, 160$ pure-group inference experiments use $r = J/2$ and heterogeneous known unary fields. The asymmetric $J = 120$ experiment also fixes the total at 60 and introduces unequal field preferences across the two modes. Positive Gaussian quadrature uses 2, 4, 8, 16, 32, 64, or 128 nodes. Target-informed positive sparse mixtures use 1, 2, 4, 8, or 16 nodes. The asymmetric control jointly optimizes locations and weights using analytic χ^2 gradients and four deterministic starts. The repaired best four-component comparator, rather than an earlier optimizer failure, is the result reported in the main text. Some larger mixtures did not declare optimizer convergence; no claim of global mixture optimality is made.

The pure-block learning study fixes $J = 120$ and evaluates four new random balanced groups. Three independently generated source streams per graph are run at each of 30 and 100 observations for each of the three source totals. Repeated streams measure synthetic sampling variation conditional on a graph. PCA centers observations within each source total before pooling them; the top $J/2$ coordinates of the leading covariance eigenvector receive positive labels. For a second factor, the next eigenvector is ranked within each dominant group to enforce balance. The target-informed mixture baseline has the true grouping and target law, whereas the compiled learner uses source estimates. The efficiency ratio combines sampling cost with population importance efficiency relative to the true target; target KL is reported separately.

D.3 Residual strength and second-factor learning

The residual study uses $J = 120$, target total 54, dominant coefficient 2.8, and second coefficients $0.25, 0.5, 1, 2, 4$. The second factor is orthogonal to the first with four equally sized signature classes. Full two-factor compilation gives exact reference probabilities. For weak residuals, target-informed two-field mixtures are optimized jointly over their field positions and positive weights. Table 4 records the entire residual-strength set for the dominant-only proposal.

D.4 Structure-learning development and fresh confirmation

The 12-fit development study uses four random structures and three streams, with totals $42, 60, 78$ each supplying 100 configurations. The second factor is balanced within each dominant group. The dominant grouping is estimated first. Spectral initialization is obtained from within-support-centered observations and projected to the allowed class. The pseudolikelihood structural control uses all feasible pair exchanges, continuous relaxation, source-only selection, and true coefficients during that stage. Every selected structure subsequently has its interaction coefficients fitted on

Table 4: Exact dominant-only proposal diagnostics across the residual-strength set. KL is $\mathrm{KL}(p\|q)$. Large importance variance at $t = 2$ is not equivalent to large KL.

Residual strength	Population ESS fraction	Target KL
0.25	.998568	.000688
0.50	.993480	.002986
1.00	.9639	0.014361
2.00	1.50×10^{-11}	.09697
4.00	1.06×10^{-31}	≈ 68.6

sources. The initial exact-swap diagnostic also used true coefficients; a follow-up alternating implementation starts from PCA and source coefficient estimates, reaching the same final structures. Mean target KL is 0.104053 for the dominant-only model, 0.052373 for projected PCA, 0.057076 for the pseudolikelihood structure, 0.047066 for exact swaps, and 0.001657 for true groups with source-fitted coefficients. Exact swaps win 9/12 and 11/12 individual comparisons against PCA and pseudolikelihood. These runs are developmental evidence.

The subsequent confirmation fixed the procedure before drawing six fresh balanced orthogonal structures (seeds 202609260800–202609260805), each with three new independent source streams. Known local weights retain the same prespecified source/target shift. The proposed procedure uses PCA initialization, exact source coefficient fitting, and at most three alternating rounds of exact swap search and coefficient fitting. It uses neither true factors nor true strengths. The pseudolikelihood control has three deterministic initializations (signed PCA, scaled PCA, and a fixed random start); it uses true interaction strengths during structural fitting and source-refits the coefficients afterward. No target outcomes select initializations or methods. The comparison rule required lower mean target KL than both projected controls and improvement on at least four of six structure-level averages against each. Both comparisons improve on six of six.

For each control, we first average the paired KL differences over the three streams of a structure. The six resulting differences define the mean and the descriptive interval $\bar{d} \pm t_{5,0.975} s_d/\sqrt{6}$. These small-sample intervals summarize between-structure variability within the generator family. They do not treat the 18 streams as independent structures. The complete confirmation used approximately 282 CPU seconds and 327 MB peak resident memory. The directly enumerated cross-model audit agrees to 1.8×10^{-15}. The plan, code, per-stream receipts, source losses, selected coefficients, and all target risks are retained together.

One selected pseudolikelihood fit initially lacked the optimizer convergence flag. A separate control check uniformly increased the same optimizer's budget for all 18 fits. All selected fits then converged; every projected structure's target KL was unchanged, so the earlier six-structure table in Appendix C and its comparisons are identical. Both receipts are retained.

E Evidence roles and reproducibility map

Known-energy cases isolate inference; pure-block fits measure group estimation at a new total; residual studies isolate model mismatch. The 12-fit study is developmental, the six-structure study tests projected controls, and the later twelve-structure study tests a discrete PL control. Each of the latter two protocols was fixed before its own instances were generated. No reserved real-data targets enter these results. Small numerical audits are correctness tests, not statistical replications.

The companion source tree records a direct mapping from each headline table to its saved receipt, with source hashes and aggregate reconstruction. Figure points are read from these receipts rather than manually transcribed. The main technical modules are a suffix coefficient compiler, group-count sampler, exact two-factor macro evaluator, and batched swap-likelihood calculator. All plotted PDF/SVG figures are vector graphics and can be regenerated from the included figure script. The official ICLR style assets are unmodified.

F LEARNING LOCAL-FIELD COEFFICIENTS: A SECONDARY ABLATION

This follow-up reuses the six structures and 18 source streams from the confirmation, so it is a secondary development experiment. Instead of supplying the numerical unary vector, it supplies three prespecified site-covariate functions and estimates their coefficients from sources. For $z_j = (j-1)/J$, define

$$f_{j,s} = \big(\cos(6\pi z_j),\ \sin(2\pi z_j),\ \cos(2\pi(z_j - s/3))\big), \quad s = 0, 1, 2.$$

The target replaces the third component with $\cos(8\pi z_j)$. Local fields are $u_{j,s} = f_{j,s}^{\top}\theta$, with generating coefficients $(0.25, 0.10, 0.07)$. Both the source and target covariate maps are specified in advance. Source likelihood jointly fits the three shared coefficients and two interaction strengths; exact swaps and continuous fitting then alternate. Thus the unknown local preferences are covariate coefficients, not unconstrained target-specific fields.

Table 5: Secondary local-field ablation on the same six structures. All rows fit interaction coefficients on sources. The bottom pair fits local-field coefficients as well, and is compared under the same information.

Local fields	Structure procedure	Mean target KL
Supplied	Spectral + exact swaps	.032677
Supplied	Projected PL, oracle strengths	.039390
Source-fitted	Spectral	.045548
Source-fitted	Spectral + exact swaps	**.037773**

Exact refinement improves on the same-information spectral fit in all six structure-level means and 14/18 individual streams. All 18 continuous fits converge; mean absolute errors of the three local-field coefficients are 0.0150, 0.0146, and 0.0107. Direct enumeration checks cross-model KL with different unary fields to 2.5×10^{-16}. This ablation supports learning shared local-field coefficients within a known feature family; its reuse of the confirmation instances is explicit.