

Scalable Inversion of Contests with Correlated Performances, Including Softmax and Multinomial Probit

Peter Cotton*

August 2026

Abstract

Multinomial probit choice probabilities over n alternatives are Gaussian orthant integrals, computed by simulation for thirty years, one expensive integral per alternative. Inversion — determining item attractiveness from a given choice probability vector — has been considered impractical for correlated contests when n is large. Both change once the choice event is treated as what it is: a one-dimensional quadrature over the winning value. For covariance drawn from a grammar covering factor, block, and hierarchical structure, a single shared survival field prices all n alternatives at once and applies the dense tie-density Jacobian, a weighted graph Laplacian, in linear work per lattice point. Forward pricing and calibration are tested at $n = 1,000,000$ and reproduce probabilities to very high accuracy; the Geweke–Hajivassiliou–Keane simulator is two hundred times slower already at $n = 200$, and its measured cost grows superquadratically while ours is linear. In the independent and factor grammars the performance distribution need not be normal. Thurstone–Mosteller model families thereby become a practical alternative to logit at modern scale.

1 Introduction

In a rather classic setup cutting across many fields, a decision maker faces n alternatives. Her choice is modeled as a stochastic contest, where each item has some inherent attractiveness (utility) to which a random noise, often Gaussian, is added. The probability of her choosing an item is synonymous with the probability that said item will have the largest total.

The races we consider in this paper come with an additional complexity because the random component for one item is assumed to be correlated with that of another. In the case of a literal race, to which our work also applies, we are, of course, asserting a covariance on performances. For instance, sprinters in a stage of the Tour de France might prefer a slower pace.

In economics and psychology, the interpretation is less literal, but the underlying model is required for the computation of counterfactuals, such as the probability that a child will choose waffles if their favorite option, pancakes, were to be removed from the breakfast menu. Increasingly, vision and text machine learning applications present new kinds of selection and routing conundrums, many with very large fields. Indeed, it could be said that the generation

*peter.cotton@microprediction.com. The reference implementation (Python), its parity-locked ports (R, Rust, JavaScript) and the seeded scripts behind every number in this paper live at github.com/microprediction/winning; this revision’s tables were produced at repository tag `paper-r10`, package version 1.4.0.

of a token is a race and, by that measure, perhaps a billion contests are being held in the time it takes to read this paragraph.

We initially consider the multinomial probit model (an example of the race we speak of) as a point of reference to prior art, even though our method does not impose an assumption of normal performance, nor any particular distribution family for that matter. Write $U_i = \mu_i + \varepsilon_i$ with $\varepsilon \sim N(0, \Sigma)$; the probability that alternative i is chosen is

$$p_i = \Pr\{U_i \geq U_j \text{ for all } j\} = \Pr\{\tilde{U} \geq 0\}, \quad \tilde{U}_j = U_i - U_j, \quad (1)$$

an $(n - 1)$ -dimensional Gaussian orthant integral. The appeal of this inherently natural generative model has been understood since Thurstone (1927). The Σ allows alternatives to be close substitutes, but even for $\Sigma = I$ this departs from the choice axiom of Luce (1959) when treated as a model for preference revealed by item removal. The axiom would mandate a simple linear renormalization of probability. But if (1) is calibrated to known p_i and then one runner is removed, the new race speaks otherwise except for very special situations (such as a symmetric contest).

Although it is out of scope, most readers would not be surprised if the disagreement between counterfactuals implied by (1) in this manner and Luce’s Choice Axiom is often in the right direction for a diverse list of applications (Cotton, 2026).¹ Practitioners in signal processing and demand estimation have had their own reasons for preferring Gaussian races to those built on fixed-location exponential performance distributions, or on the Gumbel-noise variants of (1) that look like unnatural attempts to satisfy Luce’s Choice Axiom. In any case it would be desirable to develop numerical methods that support all possibilities, to remove the evidenced temptation to select a model based on its analytical tractability, and that is precisely what we deliver herein.

The computational curse has been understood since Thurstone himself, and it bites already in his simplest case. In the original psychometrics terminology, (1) with equal discriminial dispersions and zero correlations is Thurstone’s Case V; Mosteller (1951) showed the correlations may be relaxed from zero to any common value without changing the method, and supplied the estimation and testing apparatus, which is why the family carries both names.² The integral (1) has no closed form, and estimation needs it and its derivatives at every trial value of the parameters. It is no surprise that the considerably easier logit models tend to be far more popular than probit, not only for calculations such as their obvious extension to permutation probability, but also in the very architecture of most of the world’s neural networks.

The workhorse answer to the challenge posed by (1) and calibration in particular (for which Monte Carlo is clearly problematic) is the GHK simulator of Geweke (1989), Börsch-Supan and

¹The practitioners are worth listening to. Benter (1994), whose Hong Kong betting operations by his own concession made close to a billion dollars overall (Chellel, 2018), wrote of the simple renormalization used to run probabilities down the finishing order (the formula of Harville (1973), which is the Luce axiom applied to place and show) that it “is significantly biased, and should not be used for betting purposes.” At that venue it also seems unnatural to add Gumbel noise to performance, rather than something closer to a mildly skewed normal.

²Mosteller’s relaxation is an early sighting of the identification this paper relies on throughout. With equal dispersions and common correlation, $\Sigma = \sigma^2[(1 - \rho)I + \rho\mathbf{1}\mathbf{1}^\top]$, and since $P\mathbf{1} = 0$ the common-correlation component — that component, not the whole covariance — lies in the choice-irrelevant direction: $P\Sigma P = \sigma^2(1 - \rho)P$, observationally equivalent to independent noise at variance $\sigma^2(1 - \rho)$. The two price every contest identically, which the engine reproduces to 6×10^{-17} . What this paper adds is the correlation that survives the projection.

Hajivassiliou (1993), Keane (1994) and Hajivassiliou et al. (1996). For the focal alternative i , GHK Cholesky-factors the covariance of the $(n - 1)$ utility differences $U_i - U_j$ and writes the orthant probability as a nested sequence of one-dimensional truncated normal probabilities, each conditioning on draws for the previous coordinates — the very recursion Domencich and McFadden (1975) wrote down deterministically (their eq. 4.32) and set aside as too cumbersome for estimation; GHK’s contribution was to simulate it. Averaging R such importance weights gives an unbiased, smooth estimate of one p_i at cost $O(n^2)$ per draw. The estimator made maximum simulated likelihood practical and it remains the default in textbooks (Train, 2009). Its costs are also standard: error shrinks as $O(R^{-1/2})$; the log of a simulated probability is biased at $O(1/R)$; gradients inherit the simulation noise; and a full probability vector costs n separate runs. A single likelihood contribution needs only the chosen alternative’s probability, so the n -fold cost falls not on the individual log-likelihood but on everything else estimation touches: share inversion, aggregate-share and market models, prediction and welfare across the menu, and the derivative of any of these with respect to all n utilities.

This paper takes a different route to the same object, following the tradition established in Cotton (2021). Also in keeping with that paper we will adopt once and for all a min-wins convention. Performances are to be considered golf scores, or running times where the smallest is the winner. To avoid confusion with utility we write $X_i = -U_i$. The crux is that the choice event itself is one-dimensional, by which we mean that contestant i wins when its time beats the field. The observation is as old as the subject, with one precision worth keeping. Domencich and McFadden (1975) write the general random-utility choice probability as a one-dimensional integral over the winning utility (their eq. 4.7, restated as eq. 4.26), and immediately judge it a representation rather than a method: without special structure the integrand conceals a multivariate calculation, only the Weibull case yields a convenient form, and the multiple-choice probit generalization is, in their words, “computationally intractable.” Under *independence* the integrand really is a density times a product of univariate distribution functions, and they price exactly that form for independent Cauchy shocks (their eq. 4.38) — one alternative at a time, concluding it is obtainable by “straightforward, but costly” numerical integration. So the representation, the independent one-dimensional quadrature, and even the sequential-conditioning decomposition that GHK later simulates (their eqs. 4.27–4.32, set aside as too cumbersome for iterative estimation) were all present in 1975, and all judged impractical. The independent specialization below is also standard competing-risks calculus: $G = \prod_k S_k$ is the all-causes survival and f_i/S_i the cause-specific hazard, so $p_i = \int G(f_i/S_i)$ is the classical cause-probability formula (Kalbfleisch and Prentice, 2002). Fosgerau et al. (2013) carry the companion representation — expected maximum as potential, gradient as probabilities, choice probability as an integral along the common maximum level — to general random-utility models. Universally,

$$p_i = \int f_i(x) \Pr\{X_j > x \text{ for all } j \neq i \mid X_i = x\} dx, \quad (2)$$

where f_i is the density of X_i : integrate over the winning time x , requiring every rival to come in slower. When performances are independent the conditional probability factorizes into a product of marginal survivals,

$$p_i = \int f_i(x) \prod_{j \neq i} S_j(x) dx, \quad (3)$$

with S_j the survival of X_j . Under the covariance grammars developed below the same product holds conditional on a few shared variables and is then averaged over them, so the winning-value integral stays one-dimensional and correlation adds only a low-dimensional outer integral.

For independent performances Cotton (2021) computed (3) for all n on a lattice by building the field $\sum_j \log S_j$ once and dividing each alternative out, and inverted the map from probabilities back to μ using a “multiplicity calculus” that tracked the expected number of ties conditioned on the winning score. That method is robust to reasonably pathological choices of performance distributions defined on a lattice (lumpy, gaps and so on).

The present approach trades some of that flexibility for an implicit assumption that performances are smoothly varying (in the loose sense) in order to lean more heavily on quadrature and achieve, empirically, much tighter inversion. The present paper also extends the field construction to correlated performances by conditioning, which itself is not a new idea by any means. Whenever Σ makes performances independent given a few shared variables, the field factorizes and all n probabilities cost $O(nLQ)$ for a lattice of L points and Q quadrature nodes.

To be plain about what is and is not claimed as new, since each ingredient has a history: the one-dimensional representation is Domencich and McFadden (1975), and the expected-maximum potential whose gradient is the probability vector runs through McFadden (1981) to Fosgerau et al. (2013); the common-survival field itself is classical competing-risks calculus (Kalbfleisch and Prentice, 2002), and the log-domain lattice engine that evaluates and inverts it for all n at once is Cotton (2021); that the cross-derivatives of the surplus function are tie densities forming a matrix with zero row sums — a graph Laplacian in all but name — is Müller et al. (2022); and inversion as an unconstrained convex program is Li (2018); and the fact that a Hessian–vector product can in principle be computed at the cost of a gradient is Pearlmutter (1994). The contribution here is the cavity organization that makes these compose at scale. Evaluated as the earlier literature leaves it, each of the n probabilities re-forms an $(n-1)$ -fold product, the vector costs $O(n^2L)$, and the Jacobian’s $\binom{n}{2}$ tie densities each need their own integral; that is where the representation stops being an algorithm, and it is also why Pearlmutter’s meta-theorem does not close the gap unaided — it prices a Hessian–vector product at the cost of a gradient, but differentiating the per-alternative evaluators the literature had returns exactly the $O(n^2L)$ it started with. The missing piece was the cheap object to differentiate. Building the survival field once and dividing each contestant out collapses the forward vector to $O(nL)$ (the move of Cotton 2021, on what is classically the competing-risks field; see below), and this paper shows the same cancellation collapses the *derivatives*: the dense tie-density Jacobian acts in linear work per lattice point without its edges ever being formed (§5.1), with the explicit hazard factorization additionally exposing the flux and Laplacian structure that a generic directional derivative would leave implicit. It survives correlation by conditioning, and the demonstrated million-alternative inversions run on the same field’s own-coordinate slopes as preconditioner, with the full Jacobian action exposed for Newton–Krylov solvers.

More pertinent, within such a class of covariance matrices (we will term it the factor grammar), the derivatives ride the same pass at the same cost: Jacobian–vector products and each inversion iteration are $O(nLQ)$, where again L dictates the lattice size and Q the number of quadrature points. More interesting still, the Jacobian is a weighted graph Laplacian applied as an operator and never materialized. The block and nested kernels materialize their Jacobians, and the tree’s cross-cluster block is approximate, so the matrix-free claim is the factor grammar’s, not every grammar’s. There is no per-alternative repetition as with GHK

iterating one choice at a time. The error is controlled by quadrature, deterministic for the Gauss–Hermite rules and reproducibly randomized for scrambled-Sobol nodes.

There is an obvious translation symmetry since only relative performance matters. In the language more familiar in economics, only utility contrasts enter. So strictly speaking, the inversion is a map from a probability vector back to an equivalence class of races that yield the same probabilities. The raw covariance is likewise overparameterized.

It tends to be the case in financial practice and elsewhere that structured covariance is assumed, making the clear limitation of our work less of a pragmatic concern. We do not, of course, claim a general solution for arbitrary covariance matrices, but nor do these tend to be advocated (Train, 2009). Our intent is that the grammars here cover the structured families in typical use. The first six rows of Table 1 are represented without covariance fitting.

That said, the last row is a fitted approximation developed in §6, and the utility is included in the accompanying software in anticipation of the desire to approximate the solution when an arbitrary matrix, perhaps one estimated from historical data, is supplied.

model	covariance grammar	source
Luce / softmax	independent, minimum-Gumbel base, $D = \pi^2/6$	exact identity
Thurstone–Mosteller (Mosteller, 1951)	independent, normal base	Cotton (2021)
factor multinomial probit	$VV' + \text{diag}(D)$	§4.1
teams, conferences	blocks: rank- r effect per cluster	§4.2
market plus sectors	nested: blocks $\times$ global factor	§4.2
hierarchies	tree: uniform shared effects	§4.3
dense Σ	fitted grammar + residual factors (approx.)	§6

Table 1: The covariance grammar.

Scale is a key contribution. To put it in perspective we note that on one laptop the forward computation of all- n probabilities from known utilities, assuming a rank-one factor and using the Rust kernels, takes 0.18, 2.7 and 29 seconds respectively for $n = 10^4, 10^5, 10^6$. That speed was unthinkable previously.

More impressive, in our view, is the inversion. For example at $n = 10^6$ this takes not much more than a minute for the independent case, and the rank-one factor problem can be solved at this extreme scale also, in around 20 minutes.

Comparisons die well before this scale, as far as we know, if the comparison set is taken to be those providing smoothness suitable for inversion. GHK is in that category, but we estimate its empirical complete-vector cost as superquadratic and steepening (log-log slope 2.15 from $n = 10$ to 200, 2.8 from 200 to 1000; timings at $R = 10^3$ of 0.57, 5.66 and 51.3 seconds at $n = 200, 500, 1000$, scaling by ten at $R = 10^4$). Under that measured law, and for standard one-alternative-at-a-time GHK at fixed draw count, the gap grows rapidly with n , while the shared-field pass is linear in n at fixed lattice and node budgets. We have provided in Appendix A some brave extrapolations of GHK performance to 10^4 – 10^6 alternatives.³

We proceed as follows. Section 2 places the method among the alternatives. Section 3 gives the independent lattice race, which is the same problem considered in Cotton (2021) solved slightly differently, as noted. Then Section 4 extends this across the covariance grammar. The grammar allows the set of covariance matrices to include block and tree structures with global

³See `bench.py` for reproduction.

and local factors, each one written out as a recursion. It may be of interest to those in the financial community that this structure includes that assumed in Hierarchical Risk Parity.⁴

Section 5 carries the key analytical insight and presents the Jacobian operator and its inversion. Section 6 reduces dense covariance to the grammar family, approximately, and reports accuracy per named random-correlation ensemble. Section 7 contains the benchmarks.

2 The alternatives

Deterministic quadrature. Genz (1992) transformed the orthant integral to the unit cube and integrated by adaptive and quasi-Monte Carlo rules. This is somewhat of a standard in R because `mvtnorm` chooses this path, presumably for accuracy. It certainly is accurate. The benchmark below confirms that: its answers agree with our lattice to within the sampling noise of the simulation that both are checked against. However, the disadvantage lies in what a single call returns. Genz’s method evaluates one orthant probability only, which is the win probability of one alternative. Pricing a contest requires all n of them, so the routine must be called n times, and each of those calls is itself an $(n - 1)$ -dimensional integral over the focal alternative’s difference orthant. At $n = 30$ the complete vector therefore costs about three hundred times what the lattice costs, and since the work per call also grows with n , that ratio widens as the field grows.

Frequency simulation. The crude frequency estimator of Lerman and Manski (1981) counts argmax draws. It is unbiased for the whole vector at once but not smooth in parameters, and its per-entry relative error explodes for small p_i . Measured below: at wall clock matched to the lattice it carries twelve to twenty times the total-variation error, records zero counts on tail alternatives at $n = 30$, and at ten times the budget remains a factor of two to four behind.

GHK. As described above. It is smooth in parameters but like Genz it prices one alternative per run. Despite this, GHK is seemingly the most widespread. Stata’s `cmmprobit` manual states that the likelihood evaluator “implements the Geweke–Hajivassiliou–Keane (GHK) algorithm to approximate the multivariate distribution function” (StataCorp, 2023; Gates, 2006), and the command’s entire `intmethod` menu (Hammersley, Halton, or pseudorandom point sets) selects only the sequence feeding the same simulator. The market-leading implementation offers no non-GHK method at all. R’s `mlogit` and `mvProbit` also default to GHK (Croissant, 2020), and the GHK-based `mvprobit` of Cappellari and Jenkins (2003) carried the method into applied Stata work at large.

Stern. Stern (1992) is the closest ancestor of the construction used here. He split the error into an independent normal component and a correlated remainder, observed that conditional on the remainder the alternatives are independent so that the choice probability is an analytic product of univariate normal c.d.f.s, and averaged that analytic expression over Monte Carlo draws of the remainder. The motivation was also smoothness, and similarly, Stern is testimony

⁴Although it is beyond the present scope, we mention that markets are also an example of choice probability vectors, in the sense that a dollar portfolio is allocated across stocks. The present work was motivated in part by a desire to calibrate to the market share, or to a given portfolio, under a known covariance in order to be able to re-run the same “race” with different assumptions or universe.

to a common willingness to accept a substantial per-draw cost to buy a simulator that could be differentiated. Our work differs because the conditioning variables here are a low-dimensional factor structure chosen so that the outer integral is quadrature rather than simulation: deterministic Gauss–Hermite by default, and seeded scrambled Sobol when the integrand is sharp, which replaces Monte Carlo noise with reproducible randomized-quadrature error rather than eliminating integration error outright. Either way there is no per-draw sampling noise in the returned probabilities, and, as we will discuss, the conditional problem is solved on a lattice that returns all n alternatives from a single sweep rather than one probability per run, with derivatives coming out of that same sweep.

Distributionally robust choice. A different escape from the orthant integral fixes only the first two moments of the utilities and prices the *extremal* joint distribution: the cross-moment model of Mishra et al. (2012) computes, by semidefinite optimization, the choice probabilities under the moment-feasible distribution that maximizes expected agent utility, and Ahipasaoglu et al. (2019) reformulate that computation as a convex program that scales to many alternatives. This answers a genuinely different question from ours — the worst case over a moment class rather than a specified law, with representative-agent rather than distributional semantics — so the two are complements: that family avoids ever specifying the joint distribution, while this paper prices and inverts the distribution the modeller actually specifies.

Bayesian data augmentation. Albert and Chib (1993) and McCulloch and Rossi (1994) sidestep the integral entirely by Gibbs sampling the latent utilities; Imai and van Dyk (2005) implement the marginal data-augmentation version in the MNP package, and Loaiza-Maya and Nibbling (2022) carry the Bayesian route to large choice sets by variational methods. For Bayesian inference this is a complete and often excellent answer. It is simply answering a different question than the one benchmarked here: the cost moves into the Markov chain, the likelihood surface is never formed, and choice probabilities appear only as posterior functionals, so there is no pricing call to time or to check.

Analytic approximations. Clark (1961) matched moments of pairwise maxima; Mendell and Elston (1974) and Solow (1990) approximate by sequential conditioning on first and second moments. These are fast, sometimes startlingly accurate, and carry no error control, and the benchmark below measures all three properties at once: Mendell–Elston prices the $n = 10$ vector in two milliseconds within 2×10^{-3} of the reference, then drifts to 3×10^{-2} at $n = 30$ with tail probabilities off by a factor of nearly four, and nothing in the output announces the change. The Clark-type recursion behaves alike. The family is current practice, not history: Bhat (2011) builds the MACML estimator on precisely these approximations, and the `Rprobit` package implements it (Bauer, 2024), with its asymptotics examined by Batram and Bauer (2019). Expectation propagation is the modern member of the same family. We benchmark representative sequential moment approximations through the classical members; MACML’s composite-likelihood construction around its analytic ingredient is its own estimator and is not itself benchmarked here.

Minimax tilting. Botev (2017) computes single Gaussian rectangle probabilities with bounded relative error deep into the tails, in dimensions up to roughly a thousand. It is the accuracy reference, and we use it as the referee for our own tail claims. Like GHK it prices one alternative

per run, so a whole race costs n runs, and which method is preferable depends on the question. For a single tail probability Botev’s is the right tool and this paper defers to it. For the all- n vector it is both slower and less accurate than the lattice at every size measured: at $n = 50$, 0.25 s against 0.07 s and 2.5×10^{-3} error against 2.9×10^{-5} ; at $n = 200$, 25.8 s against 0.29 s and 1.4×10^{-3} against 4.1×10^{-5} . Raising the tilting sample to $R = 10^4$ narrows the accuracy gap and widens the time one: at $n = 200$ that is 37.9 s, a hundred and thirty times the lattice’s cost and still an order of magnitude short of its error.

The reader will note the common thread. Smoothness has been, and continues to be, bought at a significant performance cost. The cost, compared to Monte Carlo, is that one alternative is computed at a time, and this cost seems to have been accepted as unavoidable. We will show that it is not, at least within the covariance grammar we specify.

We offer a different Faustian bargain that will often, but not always, be preferable. In exchange for the assumed covariance structure, which is very often assumed anyway, the user does not have to choose between speed, accuracy, scale, or smoothness. They can have all four, and very often they need them for downstream modeling tasks beyond share inversion, such as welfare analysis, game theory, and propagation of derivatives to other parts of a system (for ratings or belief networks).

3 The lattice race

The application programming interface for Cotton (2021) asked for numerical representation of performance densities on an equi-spaced lattice. The current method instead requires this assumption in the form of user-supplied functions.⁵

Emphasizing again our lowest-score-wins convention (arguably somewhat unfortunate for the special case of probit), let $X_i = \mu_i + \sigma_i \epsilon_i$ with ϵ_i i.i.d. from a standardized base density (normal, Gumbel, or any density supplied as code), and write f_i , S_i for the density and survival of X_i . From here on, μ_i is to be considered a *running time* location, the negative of the introduction’s utility location. For avoidance of any doubt, raising it makes contestant i worse and lowers p_i . Inversion returns time locations, and utility coefficients are their negation.

The algorithm evaluates (3) on a lattice $x_1 < \dots < x_L$:

1. compute $\log S_j(x_\ell)$ and $\log f_j(x_\ell)$ for all j, ℓ ;
2. accumulate the field $L(x_\ell) = \sum_j \log S_j(x_\ell)$;
3. for each i , form the integrand $f_i(x_\ell) e^{L(x_\ell) - \log S_i(x_\ell)}$, which is contestant i ’s density times everyone else’s survival, with i divided out of the shared field in log space;
4. sum times the lattice spacing, and normalize the vector.

The cost is $O(nL)$: the field is built once, not once per contestant.

3.1 Details of an adaptive lattice choice

The field is accumulated in logs. Writing $L(x) = \sum_j \log S_j(x)$ for the log survival field, evaluated once per lattice point across all contestants, each contestant’s leave-one-out product

⁵This is a crucial distinction. The lattice in the current work is only a computational device. The continuous race is more directly considered, compared to Cotton (2021) where the discrete race with explicit ties was the object under study and its relationship to the continuous race assumed.

is a subtraction rather than a division: $\prod_{j \neq i} S_j = \exp(L - \log S_i)$. This is what makes one sweep price the whole field, since the alternative is to re-form an $(n - 1)$ -fold product for each of n contestants. It is also what keeps the far tail finite. There, every S_j is small enough that $\prod_j S_j$ underflows to zero in double precision, and the division form returns 0/0 for any contestant whose own survival has underflowed; a single such point makes the integral undefined. As a sum of logarithms L is an ordinary negative number of order -10^3 , and $L - \log S_i$ is a finite difference.⁶

We are economical with the lattice points. The lattice must cover the winning time, not the spread of abilities. However hopeless a contestant is, it wins only by finishing in the range where winning times fall; thus, outside that range, the integrand is negligible. In contrast, a lattice that was stretched across the whole ability span would waste compute as most of its points lie where nothing happens.

There is a key defensive measure taken, however. Let $G(x) = 1 - \prod_j S_j(x)$, the probability that somebody has finished by x , which is the distribution function of the winning time. It is built from the caller's own base survival (rather than a normal surrogate, say), and it is made conservative at each end separately: for the lower endpoint every contestant is placed at its most favourable conditional location, and for the upper endpoint at its least favourable. Those componentwise extremes need not occur at any single factor node, which is what makes the result an envelope: wider than any one conditional race requires, and covering every conditional race represented by the retained factor nodes. Factor realizations outside the node set belong to the outer quadrature error, not to the lattice window. Solving $G(x) = \delta$ and $G(x) = 1 - \delta$ by bisection gives the two ends of the region the race is decided in. The lattice is placed there with two standard deviations of padding at each end for safety. At the achieved δ (which may be relaxed, as described next), the truncated tails are controlled separately from the bulk discretization error, and in the benchmark configurations they lie below the accuracy floors reported shortly.

But two more steps are needed before bisection can deliver what it claims.

The first is a starting interval known to contain the answer. Bisection halves an interval repeatedly and finds the point where G crosses a given level only if that point lies inside the interval it began with. A fixed starting interval does not always contain it. Nine standard deviations beyond the outermost contestant is wide enough for a Gaussian race, whose survival decays exponentially, but not for a base with polynomial tails: with Student- t performance at four degrees of freedom, the point where G falls to 10^{-12} lies 1456 standard units out.

Bisecting an interval that excludes the crossing neither fails nor warns. It converges to whichever endpoint is nearer and returns that instead. Each endpoint is therefore moved outward, doubling the distance each time, until G at the lower endpoint is at most δ and G at the upper endpoint is at least $1 - \delta$. Only then is the interval bisected.

The second requirement is that δ cannot always be honoured. The lattice has a fixed number of points, chosen by the caller, spread evenly across whatever window the quantiles ask for, so a wider window means coarser spacing. The two error sources therefore move in opposite directions: shrinking δ removes truncated tail mass but adds discretization error in the bulk, where nearly all of the probability is.

An example is instructive. For a Student- t race, $\delta = 10^{-12}$ asks for a window some 1456

⁶Survivals are floored at 10^{-300} inside every base so that $\log S$ is never $-\infty$, and the exponent is clipped to $[-745, 0]$: below -745 the exponential is zero in double precision anyway, so a hopeless contestant's contribution decays to zero rather than failing, and above 0 the value would exceed one, which no product of survivals can.

standard units wide, which at the default 257 points places them 5.7 standard units apart. That is far coarser than the scale the integrand varies on. This coarseness causes much more error than would be sustained by missing the tiny tail.

So δ is treated only as a request. If the window it asks for cannot be covered at a spacing of half the smallest performance standard deviation in the field, δ is multiplied by one hundred and the window recomputed, repeating until the spacing is affordable.⁷

We determined that it is very important to use the caller’s own survival rather than a normal surrogate. Continuing the Student- t example, a base at $\nu = 2.5$ has polynomial tails, and a normal envelope cuts the window short of them: at $n = 40$ that costs 5×10^{-3} of total variation, against 5×10^{-5} when the true survival sets the window.

Placing the lattice on the bulk is also what makes small lattices work. Thirty-three points on this window reach an accuracy that five hundred points spread across the ability span cannot, because the span window truncates the winner’s tail and its error cannot fall below about 6×10^{-11} however many points are added to it. We found that on fields with a large number of extreme longshots, the bulk window is a hundred times more accurate for the same number of points.

Another important implementation detail is that the same lattice is used to carry the derivatives. In contrast, a Jacobian evaluated on a different grid, or one that treats the lattice total as stationary when it is not, is the derivative of a neighbouring map rather than of the one that was evaluated, and this would ruin the extreme accuracy of the present implementation.

3.2 Derivatives

What a caller needs is $\partial p_i / \partial \mu_j$, in one of three strengths: the Jacobian as an operator for a full Newton method, a single row of it for an estimation score, or just the diagonal to precondition the shipped inversion. Section 5 derives what the matrix is; what matters here is how it is computed.

Prior to that we mention a lesson from the implementation, albeit one that might strike some readers as obvious: a derivative is accurate only if it differentiates the map that was *actually evaluated*, rather than the continuum integral that map approximates.

This is not a fine point. The lattice should be the very same one. Both the full Jacobian and the single-row version build their window through the routine the forward pass uses, so the two integrate over identical points.

Another potential trap lies in the fact that the diagonal should be integrated rather than assumed. Raising μ_i moves contestant i ’s own density, contributing $\partial f_i / \partial \mu_i$ to the integral. In the continuum this need never be computed, because a common shift moves no probability, so each row sums to zero and the diagonal follows from the off-diagonal entries. However, on a finite lattice that identity holds only to quadrature accuracy. For this reason, we determined that the diagonal should be computed directly.

In a similar vein, the normalization must be differentiated too. The lattice masses do not sum to one *exactly*. The returned probabilities are $p = a/T$ with $T = \mathbf{1}^\top a$. Differentiating that quotient gives $(A - p \mathbf{1}^\top A)/T$, where $A = \partial a / \partial \mu$. The correction term vanishes in the continuum, where T is constant. It does not vanish on a lattice.

⁷In the current implementation, the value finally used is reported in a warning. The window is then the stated construction at a stated δ , and a caller who wants the original one back can have it by raising the number of points.

The test is finite differences of the forward map itself: shift μ_j , re-price the field, and compare the change in p against the computed column. Write J for the Jacobian built as above, and J_0 for the continuum version, which takes its diagonal from the zero-row-sum identity and omits the normalization correction. Errors below are the largest disagreement with the finite differences, relative to the largest entry of the matrix. On a coarse 65-point Gaussian lattice J matches to 2×10^{-11} and J_0 to 1×10^{-8} ; under Student- t performance at four degrees of freedom, to 1×10^{-6} and 1×10^{-5} . On a fine Gaussian lattice, where the quadrature error is far below either figure, J and J_0 cannot be told apart.

The two directions are on a different footing, and it is worth being exact about which. Columns sum to zero identically: the quotient divides by the lattice total, so $\mathbf{1}^\top J = 0$ whatever the lattice does, at 6×10^{-17} from the coarsest grid to the finest. Rows sum to zero only as the quadrature resolves, because translation invariance is a property of the integral rather than of the sum, and the diagonal is not set to enforce it. That row residual is therefore a free measurement of lattice quality: 1×10^{-3} at 25 points, 2×10^{-10} at 65, and machine precision at the 257-point default, under either the Gaussian or the Gumbel base.

3.3 Benefits of maintaining the survival field

Two byproducts are cheap once the field is built.

The first is the density of a dead heat, which is to say the chance that i and j finish together, per unit of finishing time. It is one more sum over the same lattice. Section 5 shows that this quantity is exactly the off-diagonal entry of the derivative matrix $\partial p / \partial \mu$, so one computation answers both questions.

The second is a removal counterfactual: the probability that j wins once i is taken out of the field. In the continuum this is one more division, since deleting i simply drops S_i from the product.

But numerically, to continue our theme, it is not. The lattice was placed where the *original* winner finishes, and the survivors of a deletion need not finish there. So for example if we remove a dominant favourite, the race is decided where the original runner-up would have finished. It is entirely possible that the winner-bulk lattice would not reach that zone.⁸

Both the tie densities and the counterfactuals return $n \times n$ matrices, so their output alone is $\Omega(n^2)$. They are priced per query, not free.

4 The covariance grammar

Now we turn to the way in which we specify what covariance choices can be accommodated with extremely high accuracy, and which need to take their chances with the approximation considered later.

Of course the point is that some structures make contestants conditionally independent given a few shared variables. Without that, we have no algorithm.

In increasing order of hierarchy we consider

⁸Take $\mu = (-20, 0, 1)$ with unit variances, so the first contestant wins essentially always and the bulk window sits at about $[-29, -11]$. Remove it and the answer is $\Phi(1/\sqrt{2}) = 0.760$, writing Φ for the standard normal distribution function. Dividing the favourite out on the original grid instead leaves a total lattice mass of 3×10^{-28} where a correct integration would give one, so renormalizing it is renormalizing rounding error. Removals are therefore integrated on their own span window, with the total mass checked and a defect raised rather than normalized away.

1. a global factor,
2. factors per cluster, and
3. a tree of uniform shared effects.

Each keeps the shared field and the $O(nLQ)$ volumetrics. Each is exact given its structure; estimating that structure from data is a separate problem we do not treat here. The approximate reduction of a dense covariance to this family is deferred to §6.

4.1 Factor covariance

Let $X = \mu + Vf + \text{diag}(\sqrt{D})\epsilon$ with $f \sim N(0, I_k)$ independent of ϵ , so $\Sigma = VV' + \text{diag}(D)$. Conditional on f the performances are independent with means $\mu + Vf$, and

$$p = \sum_q w_q p^{\text{ind}}(\mu + Vf_q), \quad (4)$$

a Q -node quadrature over runs of the independent algorithm at a cost of $O(nLQ)$.

In English: average the independent race over the shared luck. The integral over f is k -dimensional, and which rule discretizes it depends on k and on the shape of the integrand.

Gauss–Hermite is the natural rule for a single Gaussian dimension. Applying it in k dimensions means taking every combination of its nodes, which costs q^k of them. At the default $q = 15$ that is 3,375 nodes at rank three and 50,625 at rank four, but 759,375 at rank five. So the product rule is used while it stays under 10^5 nodes, which in practice means rank four and below.

The other consideration is sharpness. What decides the race is not how far the factor moves a contestant but how far it moves contestants *apart*: a factor loading every contestant identically shifts the whole field and changes nothing, however large it is. The intrinsic statistic is therefore pairwise,

$$s_\Delta = \max_{i < j} \frac{\|V_i - V_j\|}{\sqrt{D_i + D_j}},$$

the shared variation available to a contrast against that contrast’s own noise, but computing it is $O(n^2k)$. The implementation first gauge-fixes the loadings, $V \leftarrow PV$ with $P = I - \frac{1}{n}\mathbf{1}\mathbf{1}^\top$ — a common loading column adds the same draw to every contestant and cannot move an argmin, so the centered matrix prices the identical race, and every downstream decision becomes invariant under $V \mapsto V + \mathbf{1}c^\top$ — and then dispatches on the linear-time bound

$$s_{\text{trigger}} = \sqrt{2} \max_i \frac{\|(PV)_i\|}{\sqrt{D_i}} \geq s_\Delta,$$

which follows from the triangle inequality on centered rows. The bound cannot miss a pair whose contrast sharpness exceeds the threshold; its false positives cost extra nodes, not error. The centered maximum is a common-shift-invariant proxy for factor-space smoothness, not a complete characterization of it, which is exactly why the dispatcher uses the conservative side of it. Nothing needs to be guessed: everything follows from V and D before any integration is done. When the trigger is large the factor draw all but decides some pairwise contest, the quantity being averaged over f is close to a step function, and a rule built for smooth integrands

converges slowly on it. The uncentered maximum row norm $\max_i \|V_i\|/\sqrt{D_i}$, tempting because it skips the centering, obeys no such inequality: centering can *increase* the largest row norm. A three-runner race with loadings $(-2.9, 0)$, $(2.9, 0.01)$, $(2.9, -0.01)$ at unit D has uncentered maximum 2.90, under the threshold, while its true pairwise sharpness is 4.10; a dispatcher trusting the raw norm stays with Gauss–Hermite there and ships a total-variation error of 6.1×10^{-3} against the analytic three-runner orthant probabilities. The test suite pins that race against its analytic answer.

A fixed rule of fifteen nodes per dimension can then miss up to five percent of the total variation, and the error is easy to overlook because the usual check does not see it. Adding lattice points does not reduce it: the lattice discretizes the finishing time, while the loss is in the average over f , so refining L returns a stable answer that is stably wrong.

Two responses follow. The node order scales with the trigger rather than staying fixed, within the rank limit above. And past a trigger of 3 at rank two or more the family changes, scrambled Sobol points replacing the product rule, since more nodes of a smooth rule is the wrong medicine for a near-step integrand. Rank one with extreme sharpness gets a third treatment, an equal-weight grid on quantiles, because Gauss–Hermite clusters its nodes near the centre and that is the wrong place when the integrand is nearly a step.

As a remark, conditioning on the factor per se is far from new. Butler and Moffitt (1982) built exactly this quadrature for the one-factor random-effects probit, and the `lpRR/slprR` interfaces of `mvtnorm` integrate over the factor dimensions of $BB' + \text{diag}(D)$ by Monte Carlo. As we noted at the outset, each of those integrals returns only one alternative’s probability, which is one reason their performance dies at scale.

The field changes that: one pass shares the survival product across all n alternatives, and §7 measures the difference against the per-alternative version of this same quadrature.

The same pass returns the own-parameter slopes $\partial p_i / \partial \mu_i$ at no extra cost (negative in this min-wins convention: raising a time mean lowers the win probability), which §5 uses to precondition the inversion.

4.2 Blocks and nested covariance

The implementation also supports groups. Assign contestant i to cluster $c(i)$ with loading $v_i \in \mathbb{R}^r$ and let

$$X_i = \mu_i + v_i' a_{c(i)} + \sqrt{D_i} \epsilon_i, \quad a_c \text{ i.i.d. } N(0, I_r). \quad (5)$$

This is a block-diagonal rank- r -plus-diagonal covariance. It can be useful for teams, conferences, and sibling products, for example. And it falls into our framework because, conditional on its own cluster’s effect, a contestant is independent of its teammates. Distinct clusters are independent altogether so the joint survival factorizes across clusters,

$$\Pr\{X_j > x \text{ for every } j\} = \prod_c G_c(x), \quad G_c(x) = \mathbb{E}_a \prod_{j \in c} S_j(x | a) = \sum_q w_q e^{S_c(x, a_q)}, \quad (6)$$

with $S_c(x, a) = \sum_{j \in c} \log S_j(x | a)$ the cluster’s conditional log-survival. To say this another way, $G_c(x)$ is just the probability that every member of cluster c survives past x , with the cluster’s shared luck integrated out.

Contestant i then wins against a “cavity” field (i removed):

$$p_i = \int h_i(x) \prod_{c \neq c(i)} G_c(x) dx, \quad h_i(x) = \sum_q w_q f_i(x | a_q) e^{S_{c(i)}(x, a_q) - \log S_i(x | a_q)}. \quad (7)$$

The term h_i couples i 's win density to its own teammates within the shared draw of $a_{c(i)}$; the cavity product supplies the rest of the field.⁹ The cost is $O(nLQ_a)$ whatever the number of clusters, because the per-cluster factors are accumulated once. Rank two is special: the conditional integrand is smooth and a tensor Gauss–Hermite rule beats scrambled Sobol at equal nodes by a wide measured margin.

One limitation of the hierarchical kernels is stated here rather than discovered by the user: their quadrature is a fixed-order rule, without the sharpness escalation of §4.1. When a cluster loading is large against the idiosyncratic noise the conditional integrand is a near-step and Gauss–Hermite fails at any order — measured at cluster sharpness 18: 5×10^{-2} of total variation against a four-million-path referee, landing on *group* shares rather than tails, with a 31-node rule moving the answer a further 8×10^{-2} rather than converging. The shipped kernels warn past sharpness 3 and direct such fields to the factor grammar, whose node family escalates automatically; porting that escalation to the hierarchical kernels is tracked work.

The hierarchical kernels place their lattice by the same envelope principle as §3: for the lower endpoint every contestant sits at its most favourable retained conditional location, for the upper at its least favourable, and the idiosyncratic scale alone sets the bisection. The obvious shortcut — an independent-marginal proxy using each runner's total standard deviation at its unconditional location — fails exactly where these kernels are interesting. On a 400-runner cluster at correlation 0.99 the proxy's window dropped 28 percent of the winner mass, asymmetrically across two loading groups, and silent normalization returned group shares of 0.68/0.32 where the symmetry of the construction forces 0.50/0.50. The raw lattice mass is therefore treated as a diagnostic rather than a nuisance: the kernels check it and raise an error on a material defect instead of normalizing it away. That check promptly earned its keep by catching an unrelated slip: the tree passes had been ordered by accumulated strength as a proxy for depth, which ties when strengths are zero — precisely the trees a linkage induces — and a tied order visits children before their parents, pricing a six-leaf zero-strength tree at raw mass 3.0.

Nested. Add one global factor with couplings g_i and a dial $\gamma \in [0, 1]$: conditional on the global draw f_q the model is a block race at shifted abilities, so $p = \sum_q w_q p_{\text{block}}(\mu + \gamma g f_q)$, a finite mixture of block races. At $\gamma = 0$ the clusters are isolated, at $\gamma = 1$ fully coupled, and the covariance contribution scales as $\gamma^2 g g'$.

4.3 Trees

Let a rooted tree have the leaf clusters at its bottom and internal nodes t above, and give node t a scalar effect $b_t \sim N(0, 1)$ applied with uniform strength λ_t to every leaf beneath it:

$$X_i = \mu_i + v_i a_{c(i)} + \sum_{t \in \text{anc}(c(i))} \lambda_t b_t + \sqrt{D_i} \epsilon_i, \quad (8)$$

⁹Here we borrow the “cavity” terminology from statistical physics, because it is reminiscent of computing the influence on one site by deleting it, asking what the remaining system does in the hole left behind, and then reinserting it. A similar construction appears as the cavity distribution in belief propagation. Note that here the deletion is quite literal and $\prod_{j \neq i} S_j$ is the field with contestant i removed. Readers from econometrics may prefer *leave-one-out*. The closest familiar object is the inclusive value of nested logit, an aggregate over alternatives that enters each alternative's own probability, with the difference that the focal alternative is excluded from the aggregate here rather than included. *Field*, likewise, is physics for $\sum_j \log S_j$, the single aggregate quantity every contestant is priced against, and the reason one pass serves all n of them.

with the leaf-cluster construction of §4.2 restricted to scalar loadings, $v_i \in \mathbb{R}$ and $a_c \sim N(0, 1)$: the shipped tree kernels take rank-one leaf-cluster effects, where the block kernels support rank r , and they refuse a rank- r leaf loading rather than price it. The implied covariance is hierarchical: two contestants share the variance of exactly their common ancestors. The root’s effect is common to every contestant, hence choice-irrelevant, and the pricing kernel gauge-fixes it to zero. A constructor that reports a covariance may still carry a root variance so that the raw matrix matches a stated target; the prices are the same either way, and Remark 1 uses exactly that freedom. The uniform strength is the crucial restriction, because a shared effect that hits every subtree member equally moves the whole subtree’s field by a translation of its argument. Messages therefore remain functions of one variable. Per-leaf loadings on internal effects would force two-dimensional message tables.

The algorithm is two passes over the same lattice.

Upward. Each leaf cluster carries its block factor $G_c(x) = \sum_q w_q e^{S_c(x, a_q)}$ from (6). Each internal node, visited deepest first, combines its children under its own effect:

$$G_t(x) = \sum_q w_q \prod_{c \in \text{children}(t)} G_c(x + \lambda_t a_q). \quad (9)$$

In English: $G_t(x)$ is the probability that every leaf below t survives past x , with t ’s effect integrated by quadrature and each child’s message shifted because the effect moves that child’s whole subtree at once. Shifted evaluations are linear interpolations on the lattice.

Downward. The root’s cavity is one. Each node, visited shallowest first, receives the field of everything outside its subtree:

$$R_t(x) = \left[\sum_q w_q R_{\text{parent}(t)}(x - \lambda_{\text{parent}(t)} a_q) \right] \prod_{s \in \text{siblings}(t)} G_s(x). \quad (10)$$

The sibling messages sit outside the parent-factor quadrature. This placement is correct but not obvious: the parent effect shifts the candidate and every sibling subtree by the same amount, so it cancels from every candidate-versus-sibling comparison; only the cavity from outside the parent must be shifted and averaged. Contestant i then wins against its own leaf cavity, $p_i = \int h_i(x) R_{c(i)}(x) dx$ with h_i as in (7). The cost is $O(nLQ)$ plus $O(LQ)$ per tree node.

A depth-one tree with zero strengths reproduces the rank-one block race to machine precision, which is tested. Against a 2^{22} -path common-random-numbers referee the recursion sits at the simulation noise floor on every resolvable entry at depths one through four (root-mean-square z between 0.75 and 1.1). Entries below the referee’s resolution agree with minimax tilting (Botev, 2017) to 0.3 percent relative at probabilities down to 7×10^{-8} , which is within the tilting estimate’s own error. Given adequate factor quadrature, tail accuracy is carried by the lattice, and the grammar kernels price tails that simulation does not resolve.

Remark 1 (A connection to top-down portfolio construction). *Long-only portfolio weights can be read as choice probabilities, and we make the further connection to dendrogram-inspired wealth bisection (the hierarchical risk parity construction of López de Prado (2016)). There is a tree race whose implied correlation is exactly the floored cophenetic correlation of a dendrogram:*

for a linkage built from the correlation distance $d_{ij} = \sqrt{(1 - \rho_{ij})/2}$, invert the distance transform at each merge height and floor at zero. Give internal node t strength $\lambda_t^2 = \rho_t - \rho_{\text{parent}(t)}$, with $\rho_t = \max(1 - 2h_t^2, 0)$ at merge height h_t , and leaf i idiosyncratic variance $1 - \rho_{\text{first}(i)}$.

The floor at zero is not optional. Shared effects contribute nonnegative correlation, so a tree race cannot represent a negative cophenetic correlation, and dropping the floor pushes the other implied correlations above one. The shipped `from_linkage` constructor keeps the root at its cophenetic value, which reproduces the floored cophenetic matrix exactly; this is the covariance-reporting choice of §4.3, and the pricing kernel ignores the common component regardless. Setting the root effect to zero instead gives the same choices, since a common effect cannot move an argmin, but different raw correlations.

Concentration limits stated on a dendrogram therefore become contest inversions. We do not pursue the application here.

5 Jacobians, tie densities, inversion

We now arrive at the central object of the algorithm: the Jacobian. Its structure has prior art we were unaware of when this work was done: Müller et al. (2022), differentiating the Williams–Daly–Zachary gradient identity $\partial E / \partial u_i = \Pr\{i \text{ wins}\}$ for general joint laws, derive the cross-Hessian as the density that two alternatives tie for the maximum, and observe that the resulting matrix is symmetric with signed off-diagonals and zero row sums — the Laplacian class in all but name, though they draw no graph and build no algorithm from it, their object being convexity moduli for prox-functions, and their cross-derivative remains an $(n - 2)$ -fold integral of the joint density. The face parametrization below reduces the same object to a one-dimensional integral of a pair density against the conditional survival of the field, which is the form a lattice can price. We arrived at the same structure independently, by a different route, and the route is what we keep: a derivation of the structure as a literal probability flux through the boundary between winning regions, which delivers the formula, the symmetry, the conservation law and the graph structure in one geometric picture; and, on the computational side, §5.1’s identity letting the dense matrix act in linear work. The geometry first.

First, an outcome of the race is a vector x of finishing values, one per contestant. Contestant i wins on the set

$$R_i = \{x : x_i \leq x_k \text{ for every } k\},$$

a convex cone: the outcomes in which nobody beats i . These cones tile the space, one per contestant, and each win probability is the mass the joint density puts on its own cone,

$$p_i(\mu) = \int_{R_i} q_\mu, \quad q_\mu(x) = q_0(x - \mu).$$

Where two cones meet, two contestants dead-heat ahead of the field. So we refer to the shared boundary

$$F_{ij} = \{x : x_i = x_j \leq x_k, \ k \neq i, j\}$$

as the photo-finish face between i and j .

Now differentiate. Abilities enter the density only as a translation, so raising μ_j slides the density rather than reshaping it, and sliding the density one way is the same as sliding the

coordinate the other: $\partial q_\mu / \partial \mu_j = -\partial q_\mu / \partial x_j$. The region R_i does not depend on μ at all, so for $j \neq i$ the whole derivative falls on the integrand,

$$\frac{\partial p_i}{\partial \mu_j} = - \int_{R_i} \frac{\partial q_\mu}{\partial x_j} dx.$$

We recognize this as a derivative integrated over a region, so the divergence theorem applies to the vector field $q_\mu e_j$. This lets us compute over the boundary instead of the volume:

$$\frac{\partial p_i}{\partial \mu_j} = - \int_{\partial R_i} q_\mu n_j dS,$$

with n the outward unit normal.

However for this we take q_0 continuously differentiable with q_0 and ∇q_0 integrable and $|x|^{n-1} q_0(x) \rightarrow 0$, conditions any Gaussian law satisfies outright. Integrability alone would license neither differentiating under the integral sign nor discarding the flux at infinity. With them the contribution over the sphere at infinity vanishes, and only the faces of R_i remain.

The observation: those faces are the photo-finish surfaces F_{ik} , one for each rival k , and on F_{ik} the outward normal is $(e_i - e_k)/\sqrt{2}$. Its j -th component is zero unless $k = j$. Every face therefore drops out except F_{ij} . What is left is the statement the proposition makes precise: slowing j pushes probability out of j 's winning region into i 's, and all of it crosses at the surface where the two dead-heat.

Proposition 1 (Photo-finish derivatives are a boundary flux). *For $j \neq i$,*

$$\frac{\partial p_i}{\partial \mu_j} = \frac{1}{\sqrt{2}} \int_{F_{ij}} q_\mu dS \geq 0, \quad (11)$$

the probability flux across the surface on which i and j dead-heat ahead of the field. Parametrizing that face by the common value $x_i = x_j = t$ contributes a surface factor $\sqrt{2}$ which cancels, giving, with φ_{ij} the joint density of (X_i, X_j) ,

$$\frac{\partial p_i}{\partial \mu_j} = \int \varphi_{ij}(t, t) \Pr\{X_k > t \ \forall k \neq i, j \mid X_i = X_j = t\} dt, \quad (12)$$

and under independence

$$\frac{\partial p_i}{\partial \mu_j} = \int f_i(x) f_j(x) \prod_{k \neq i, j} S_k(x) dx. \quad (13)$$

Under conditional independence the same factorization holds node by node and is averaged over the shared variables. Each row of the Jacobian sums to zero.

Everything else follows from the geometry. Symmetry is $F_{ij} = F_{ji}$: one surface, one number, so $w_{ij} = \partial p_i / \partial \mu_j$ is symmetric and nonnegative. The zero row sum is translation invariance, a conservation law: what leaves one cone enters its neighbours. Writing B for the oriented incidence matrix of the photo-finish graph and W_e for the diagonal of edge conductances w_{ij} ,

$$J = -B^\top W_e B, \quad (Jh)_i = \sum_{j \neq i} w_{ij} (h_j - h_i),$$

the discrete analogue of $-\text{div}(a\nabla h)$: Bh takes differences across photo-finish boundaries, $W_e Bh$ is the flux across them, $B^\top W_e Bh$ the net flow into each winning region. The mass-one identity is the same theorem in dimension one, since $\sum_i f_i \prod_{j \neq i} S_j = \frac{d}{dt}(1 - \prod_j S_j)$ integrates to one; this is why the implementation’s unnormalized mass checks are diagnostics rather than cosmetics, a material defect meaning the lattice missed the region, not that normalization was needed.

5.1 Jacobian computation

Scalability rests on never materializing the matrix. Full Newton needs the Jacobian only as an operator applied to a vector; a likelihood score needs one row of it; and the shipped inversion below needs only the own-coordinate preconditioning slopes.

The identity the scale rests on. Per factor node, write $A_i(x) = f_i(x)/S_i(x)$ for contestant i ’s hazard and $G(x) = \prod_k S_k(x)$ for the field. The tie density factors through the field,

$$w_{ij} = \int G(x) A_i(x) A_j(x) dx, \quad j \neq i,$$

so the apparently dense Laplacian is a continuous sum of rank-one contributions, and its action collapses: substituting into $(Jh)_i = \sum_{j \neq i} w_{ij}(h_j - h_i)$, the $j = i$ terms cancel between the two sums, leaving

$$(Jh)_i = \int G(x) A_i(x) \left[\sum_j A_j(x) h_j - h_i \sum_j A_j(x) \right] dx.$$

The two inner sums are shared across every i , so the whole product costs $O(nL)$ per node — the same order as pricing — rather than the $O(n^2L)$ of assembling edges. This, more than the Laplacian structure itself (which Müller et al. (2022) already had), is what turns the tie-density Hessian into a scalable object: a graph with $\binom{n}{2}$ potential edges is applied without ever being constructed.

What gets computed. Both a single entry and a Jacobian–vector product come out of the same field pass that produced the probabilities, by the identity above (in the log domain, with hazards clipped as §3 describes). Forming all n^2 entries is possible, but it is a choice with $\Omega(n^2)$ cost in output alone, and nothing downstream requires it.

Four derivatives that are not the same. In keeping with our previous remarks we delineate four objects. The first is the continuum derivative, equation (13), the exact derivative of the integral that defines p . The second is that formula evaluated by quadrature on a lattice, with the diagonal imposed from the zero-row-sum identity. The third, call it the grid derivative, is the exact derivative of the normalized rectangle sum *holding the selected grid fixed*: with unnormalized masses a , total $T = \mathbf{1}^\top a$ and $A = \partial a / \partial \mu$, it is $(A - p \mathbf{1}^\top A) / T$. The fourth is the exact derivative of the full adaptive map, which adds a grid-motion term, since the window itself depends on μ .

The factor implementation returns the third. It differentiates the rectangle sum itself, then applies the normalization quotient: a directional derivative v of the unnormalized masses is returned as $(v - p \mathbf{1}^\top v) / T$. It does not differentiate the adaptive window, and the residual measured below is exactly that omission.

That quotient is critical. Without it, the returned directional derivative disagrees with central finite differences of the returned map by 2×10^{-3} at $L = 25$ lattice points. With it the two agree to 3×10^{-11} from $L = 101$ upward, which is the production range.

At $L = 25$ a residual of 6×10^{-4} survives, and it is not a defect in the quotient. It is the grid-motion term, the difference between the third object and the fourth: a finite difference recomputes the window at each perturbed μ and therefore sees a contribution the fixed-grid derivative does not contain. It falls below the agreement figure by $L = 101$, so in the production range the grid derivative serves as the derivative of the adaptive map to the accuracy reported.

The materialized Jacobians of the block, nested and tree kernels are the *second* object, not the third: continuum tie densities evaluated by quadrature with the diagonal imposed, ignoring both the normalization quotient and the window. They are consistent estimates of the continuum derivative, and they are not exact derivatives of the normalized discrete maps they ship beside; a caller who needs the latter (an optimizer, say) should difference the forward map, which is what the polish stage does to restore feasibility when the approximation shows.

The continuum form is exposed alongside the grid form, because it is symmetric by construction and so can be used as a preconditioner without further checking. The grid form is symmetric only to the accuracy of its row sums, for the same reason: at 25 points it is off by 3×10^{-4} and at 65 by 7×10^{-11} , but at the 257-point default the two forms agree to machine precision, so the distinction is one of guarantee rather than of observed values in production.

Keeping the pair term finite. The photo-finish integrand is factored as $[f_i e^{L - \log S_i}] \times [f_j / S_j]$. The first factor is bounded by a density. The second is a hazard rate, which grows linearly under the normal base and exponentially under the Gumbel base; the exponent clip absorbs either. The symmetric square-root split, which looks more natural, overflows wherever survivals vanish.

Across the grammar. The block Jacobian is exact for rank-one cluster loadings, with the within-cluster term computed under the cavity; the rank- r forward kernel has no Jacobian yet, and the call refuses rather than mis-broadcasting. The nested Jacobian is the mixture of block Jacobians. The tree Jacobian is exact within a cluster and Gram-approximate across clusters. That approximation is safe for Newton because the residual is always computed from the exact forward map, so an approximate Jacobian can slow convergence but cannot move the answer it converges to. Callers for whom feasibility is critical fall back to finite differences.

5.2 Inversion

The inversion rests on a convex program.

Theorem 1 (The inverse exists and is unique on contrasts). *Fix a covariance of full support (any member of the grammar with every $D_i > 0$) and let $W(\mu) = \mathbb{E} \min_i (\mu_i + \epsilon_i)$, $\epsilon \sim N(0, \Sigma)$. Then W is concave and differentiable with $\nabla W(\mu) = p(\mu)$, and its Hessian is $-L(\mu)$, the negative of the graph Laplacian whose edge weights are the correlated tie densities (12), reducing to (13) under independence. Every tie density is strictly positive, so W is strictly concave on the contrast space $\mathbf{1}^\perp$. For any target with $p_i^* > 0$ and $\sum_i p_i^* = 1$, the program*

$$\max_{\mu} W(\mu) - \langle p^*, \mu \rangle$$

is constant along $\mathbf{1}$, strictly concave and coercive on $\mathbf{1}^\perp$, and its unique mean-zero maximiser is the inverse: $p(\mu^*) = p^*$. If some $p_i^* = 0$ the supremum is approached only as the contrast $\mu_i - \min_{j: p_j^* > 0} \mu_j \rightarrow +\infty$ (the raw coordinate statement depends on the gauge); no finite inverse exists, and any floored target yields a finite point on that diverging contrast.

Proof. A minimum of affine functions of μ is concave and expectation preserves this. Ties have measure zero under full support, so the minimiser is almost surely unique, and Danskin's theorem (Danskin, 1966) gives $\partial W / \partial \mu_i = \Pr\{i \text{ wins}\}$ — the identity known in discrete choice as the Williams–Daly–Zachary theorem (McFadden, 1981).¹⁰

The Hessian is given by Proposition 1, which assumed no independence: its off-diagonals are the boundary fluxes (12) under the full correlated law, and its rows sum to zero by the same translation invariance $W(\mu + c\mathbf{1}) = W(\mu) + c$ that fixes the diagonal without further computation. The Hessian is therefore $-L(\mu)$ with L the weighted graph Laplacian of the photo-finish graph; under full support ($D_i > 0$, so every pairwise difference has positive variance) each edge weight is strictly positive, the tie graph is complete, and $L \succ 0$ on $\mathbf{1}^\perp$: W is strictly concave there.

Coercivity follows from the elementary bound $W(\mu) \leq \min_i \mu_i$: for each i , $\min_k (\mu_k + \epsilon_k) \leq \mu_i + \epsilon_i$, and $\mathbb{E}\epsilon_i = 0$. For a nonconstant unit contrast d and interior target, $\langle p^*, d \rangle > \min_i d_i$, because p^* puts positive mass on a coordinate where d exceeds its minimum; hence

$$W(td) - t\langle p^*, d \rangle \leq t(\min_i d_i - \langle p^*, d \rangle) \rightarrow -\infty$$

linearly in t . The decay is uniform, not merely raywise: $d \mapsto \langle p^*, d \rangle - \min_i d_i$ is continuous and strictly positive on the compact unit sphere of $\mathbf{1}^\perp$, so it attains a positive minimum κ , and the displayed inequality at $t = \|\mu\|$ gives $F(\mu) \leq -\kappa\|\mu\|$ for every $\mu \in \mathbf{1}^\perp$, writing F for the objective, which is coercivity for arbitrary diverging sequences. (The bound is on F itself, not on F relative to $F(0)$: the objective may climb toward its maximizer before the linear decay takes over.) Constancy along $\mathbf{1}$ is translation invariance again, exactly when the target sums to one. A strictly concave coercive function on $\mathbf{1}^\perp$ attains a unique maximum, where the gradient vanishes: $p(\mu^*) = p^*$.

For the boundary claim, suppose $p_i^* = 0$ for i outside $S = \{j : p_j^* > 0\}$. Under full support $p_i(\mu) > 0$ at every finite μ , so no finite point inverts the target. Invert the target restricted to S , then send each excluded $\mu_i \rightarrow +\infty$ holding the S -contrasts fixed: by dominated convergence the race converges to the restricted race and the objective to its restricted optimum, so the supremum is approached along exactly those diverging contrasts. Conversely, on any maximizing sequence an excluded contrast that remained bounded would admit, after gauge fixing, a convergent subsequence whose limit gives alternative i strictly positive probability, and the objective, continuous in μ , would sit strictly below the restricted optimum there — contradicting maximality. $\square$

We remark on the connection to economics. We are considering the Legendre dual of the social surplus function of McFadden (1981). Share inversion has frequently been viewed as

¹⁰Danskin's theorem differentiates a minimum over a parameter. For $W(\mu) = \mathbb{E} \min_i (\mu_i + \epsilon_i)$ the naive move, differentiating inside the expectation, is illegitimate because the minimum is not differentiable where two contestants tie. The theorem says the derivative exists anyway whenever the minimiser is unique, and equals the derivative of the selected branch: here $\partial / \partial \mu_i$ of $\mu_i + \epsilon_i$, which is one, on the event that i is the minimiser and zero otherwise. Taking expectations gives the win probability. Full support is what supplies uniqueness, since ties then have measure zero; without it W has kinks and only a subdifferential. This is the step that makes p a gradient, and hence the inversion a convex program rather than a root-find.

a convex problem: Berry (1994) inverts market shares in logit-family demand, Galichon and Salanié (2022) identify the general random-utility inversion with optimal transport, Chiong et al. (2016) compute it by linear programming in the discrete case, and Müller et al. (2022) state the inversion as the subdifferential of the conjugate surplus, $u \in \partial E^*(p)$, en route to their prox-functions.

Closest is Li (2018), who for arbitrary non-atomic bases writes the inversion as an unconstrained convex minimization whose objective gradient is the share map, with globally convergent superlinear methods. That is a formulation of the optimizer, but without a fast evaluator of the shares and their Jacobian at large n , which is what we supply. The shared field is that evaluator, so the two are complements. What the field representation adds is not the convex program but its oracles: the shipped solver preconditions with the field’s own-coordinate slopes, and the Hessian — the tie-density Laplacian of the same pass — is available as a matrix-free operator at $O(nLQ)$ per application for Newton–Krylov methods.

As noted in passing earlier, an invariance applies to the covariance and not just to translation. Choice probabilities are invariant to a common utility shock, so the model depends on Σ only through $P\Sigma P$ with $P = I - \mathbf{1}\mathbf{1}^\top/n$: a factor proportional to $\mathbf{1}$, or a tree-root effect applied to every leaf, is choice-irrelevant. The right fitting objective for a dense covariance is therefore the projected residual,

$$\min_{V, D \geq 0} \|P(\Sigma - VV' - \text{diag } D)P\|_F^2,$$

not a fit of $VV' + \text{diag}(D)$ to $P\Sigma P$ (the projection of a rank-plus-diagonal matrix is no longer rank-plus-diagonal, so the second problem has some manufactured error compared to the first).

The reference implementation minimizes that objective in `factor_model_projected`, alternating two steps. The V -step takes the top k eigendirections of the current residual in the contrast basis. The D -step solves for the diagonal exactly, subject to a lower bound, from the normal equations

$$(P \circ P) d = \text{diag}(P(\Sigma - VV')P),$$

whose Gram matrix is closed form, so the step is $O(n)$ rather than a least-squares solve (§6).

Everything downstream works from projected residuals, so a matrix already of the form $VV' + \text{diag}(D)$ passes through whenever its factor rank lies within the fitted grammar’s budget — up to the common-shock equivalence, factor-quadrature error, and a diagonal floor of 10^{-6} of each runner’s own variance.¹¹

The implemented inversion (`abilities_from_race`) runs as follows.

¹¹Where the introduction speaks of an identified class, it means the restriction that is choice-relevant and computable on $\mathbf{1}^\perp$: choices depend on Σ only through $P\Sigma P$, an observational-equivalence statement, not a claim about what an estimation design can recover. Covariances outside the grammar are perfectly good models; this grammar simply does not price them.

Inversion (all grammars; min-wins). Given a target p^* with positive entries summing to one, set $t = \log p^*$ and initialize $\mu^0 = -(t - \bar{t})/2$. Repeat until $\max_i |r_i| < 10^{-8}$:

1. one field pass: $\hat{p} = p(\mu)$, and negative preconditioning slopes $\tilde{s}_i$ (factor grammar: the own-coordinate grid slopes $\partial p_i / \partial \mu_i$ of the same pass; block/nested/tree: the own-slopes of an independent race at matched total variances, one extra $O(nL)$ pass — an approximation used only to precondition, never to form the residual);
2. log residual $r_i = \log \hat{p}_i - t_i$; log-domain slope $d_i = \min(\tilde{s}_i / \hat{p}_i, -10^{-6})$;
3. damped, clipped preconditioned step $\mu_i \leftarrow \mu_i - \text{clip}(\alpha r_i / d_i, \pm 2)$, then recenter $\mu \leftarrow \mu - \bar{\mu}$.

Damping: $\alpha = 1$ (0.7 at $n = 2$, where the undamped update on the K_2 photo-finish graph is a two-cycle); for the structure grammars $\alpha = 0.7$, halved whenever the residual fails to shrink.

Each sweep is a diagonally preconditioned residual iteration on the log-probability residual $r_i = \log p_i(\mu) - \log p_i^*$.¹² A full Newton step would solve a linear system in the Jacobian; here each coordinate is instead divided by a preconditioning slope $\tilde{s}_i$, which for the factor grammar is the own-coordinate slope the forward pass already returns, and for the other grammars an independent-race approximation to it. The residual, by contrast, always comes from the exact structured forward map, so the preconditioner does not change the root equation; it can affect convergence and the finite-tolerance point returned. One field pass per iteration, no linear solve. Working in logs rather than in probabilities matters because win probabilities span many orders of magnitude, and an absolute residual would effectively ignore everything but the front of the field. A step clip and a recentering keep the iterate in the contrast space.

Theorem 1 says the inverse exists and is unique for the Gaussian grammars, which is where it is stated. For a general base the same proof goes through when the location family has finite first moments, an everywhere-positive sufficiently regular density and no atoms: concavity and coercivity of W use nothing Gaussian, and strict concavity needs only positive tie densities, which full support supplies. We have not written that generalization out formally, so for non-normal bases the uniqueness claim should be read as proved under those conditions rather than universally. The theorem also does not say that this clipped iteration finds the inverse. The solver’s convergence as implemented stands on the empirical evidence, which is strong on the benchmark families but not universal: we can construct strongly correlated fields on which the damped iteration stalls at a residual near 10^{-4} instead of converging.¹³

¹²In handicapper’s terms, for a million runners: you know the win probability you want for each runner and must adjust handicaps until the model produces it. Each pass of the reported solver computes every runner’s current win probability and every runner’s *own* slope — if only this runner slows slightly, how much does its own probability move? — then adjusts each runner by its personal sensitivity, re-prices the whole race, and repeats. Interactions are not ignored (every re-pricing includes them); they are just not consulted when choosing each step. The Jv oracle is the more powerful tool: propose changing *all* abilities at once by v and it returns the resulting change in *all* win probabilities, without constructing the million-by-million Jacobian’s 10^{12} entries. A Newton–Krylov solver would repeatedly ask it what coordinated change removes the current errors most efficiently. That more sophisticated solver is not what produced the reported million-runner result; the own-slope iteration did, and making Newton–Krylov beat it is future solver work.

¹³A matrix-free Newton–Krylov solver using the exact grid Jacobian–vector product is provided and measured (the committed `exp23`, three rounds). Naively assembled it loses badly to the damped diagonal iteration — 387 seconds without converging against 3.7 seconds converged at $n = 200$. Assembled properly — log-residual system

We tested the round trip at $n = 400$, with twenty clusters and a tree of depth three (the committed `bench.py invert`). Timing for independent was 0.04 s, tree 0.37 s, and blocks 0.67 s. A factor rank-two took 1.7 s and nested 3.7 s. In each case the maximum log-probability residual was below 10^{-8} .

The implementation does not hide errors. A zero or negative entry raises rather than being quietly repaired, since by Theorem 1 it has no finite inverse. Flooring small entries is available but must be requested by the user. For an entry whose target is exactly zero, the theorem says every finite-floor solution is a finite point on a contrast that diverges as the floor is removed, so the floored answer understates that contrast in the direction the theorem indicates. For entries that are merely small and positive, reading the floored-and-renormalized solution as a coordinatewise bound on the true contrasts would need a monotonicity argument we do not make here.¹⁴

Every grammar gets that same contract. The target is validated once, before the call splits by structure, and all five paths return through a single exit, so a block or tree inversion reports floored entries, iteration count and convergence exactly as the factor path does.

Structure matters at inversion time, and not by a little. Inverting block-generated probabilities under an assumed independent model mislocates abilities by up to three tenths of the field’s spread: at $n = 200$ with thirty clusters and within-cluster correlation 0.65, the largest mislocation is 2.49 against a spread of 8.24, with a median of 0.34.

6 Approximate results for dense covariance

We ship a fitting procedure as follows for the general case. It comes with several caveats.

We use the following notation. Let $s_i = \sqrt{\Sigma_{ii}}$, $S = \text{diag}(s)$, and $C = S^{-1}\Sigma S^{-1}$ the correlation matrix. Every fit, residual and closing solve below is carried out in Σ . Correlations appear only once, as the distance used for clustering in stage 3.

That choice is apparently forced, as we discovered. The tempting alternative is to standardize to C , fit there, and restore the scales afterwards, which would let the fitter work on a matrix with unit diagonal. It does not work, and the reason is worth spelling out. Choices depend on Σ only through $P\Sigma P$, so the residual that matters after restoration is $PS(C - M)SP$, not $P(C - M)P$, and those differ because P does not commute with S unless every scale is equal. Put another way: the direction choices ignore is the constant vector in covariance coordinates, but $S^{-1}\mathbf{1}$ in correlation coordinates. Fitting in correlation space with the ordinary projection removes the wrong direction.¹⁵

symmetrized by the $\sqrt{p}$ scaling, the integration-by-parts form of the product, conjugate gradients preconditioned by the own-slopes, null vector projected out, trust clip and a true-residual acceptance test — it does converge on the benchmark fields (9.6×10^{-10} , at roughly sixty times the wall clock), and on the constructed stall case it diverges where the diagonal iteration stalls. So the diagonal iteration is the production solver, and a robust Newton–Krylov remains solver engineering rather than a capability gap in the oracle. The nearest proven result we know for a problem of this shape is the damped Newton method of Kitagawa et al. (2019) for semi-discrete optimal transport: prescribe the mass each Laguerre cell must capture and adjust one weight per cell, with a Hessian assembled from source mass on shared cell boundaries much as our Jacobian is assembled from tie density on shared win-region boundaries. They prove global linear convergence under regularity conditions on the cost and source; Lévy et al. (2021) run the method at ten million cells. Whether a race-market analogue of their conditions yields a convergence proof here is open.

¹⁴At time of writing the diagnostics name which entries were floored, whether the iteration converged, and the residual it reached; non-convergence warns.

¹⁵This really matters. Take $n = 2$, C equicorrelated with $\rho = 0.9$, and scales $s = (1, 2)$. With two alternatives

A single global rescaling of Σ for conditioning is harmless; it is unequal scales that break the commutation. The test suite pins this two-alternative example, and a round trip on a covariance built as $VV' + \text{diag}(D)$ exactly, with idiosyncratic variances drawn across a hundredfold range: pricing that matrix through the dense front door must reproduce the race obtained from the (V, D) it was built from. Nothing there is approximate, so any disagreement is the coordinate choice rather than the grammar.

The stages, in order:

1. Global factors: the quotient fit of §5 (`factor_model_projected`), $k = 3$ throughout, returning (V, D_0) by alternation. Each half-step is optimal in its own block, so in exact arithmetic the projected objective descends monotonically to a coordinatewise-stationary point. Above $n = 800$ the V -step is subspace iteration rather than a full eigendecomposition and its subspace is only approximate, so descent is enforced instead of assumed: the objective is evaluated each sweep at $O(n^2)$, a sweep that fails to improve it is discarded, and the alternation returns the best iterate. The alternation is nonconvex and its default start can stall on easy inputs (an exactly in-grammar matrix at $n = 12$ with variances spread across two decades sat at objective 0.88 where zero was attainable), so when the first start's objective is materially above zero, two further deterministic starts are tried — diagonal-heavy and eigen-residual — and the best kept; each rescues that case to 10^{-13} .
2. The working residual is then projected once, $R = P(\Sigma - VV' - \text{diag } D_0)P$: the raw residual still contains the choice-irrelevant common component the quotient fit rightly ignored, and letting later stages chase it manufactures projected error (measurably: the raw-residual variant distorts an input already of the form $VV' + \text{diag}(D)$ by 1.7×10^{-2} of total variation; the projected pipeline returns it at the 2×10^{-4} node-noise floor).
3. Blocks: average-linkage clustering on $\sqrt{(1 - C)/2}$ cut to at most 20 clusters; for each cluster, the leading eigenvector of the within-cluster block of R with its diagonal zeroed, one rank-one loading per cluster (clusters are disjoint).
4. Residual promotion: the top $m = 5$ eigencomponents of what remains of R , diagonal zeroed, appended as further factor columns; negative eigenvalues are not promoted; numerically dead columns are dropped.
5. The closing diagonal solves the projected problem's own normal equations, $(P \circ P)d = \text{diag}(P(\Sigma - V_{\text{all}}V'_{\text{all}})P)$ subject to $d_i \geq \ell_i$, where the floor is relative to each runner's own variance, $\ell_i = 10^{-6} \Sigma_{ii}$. Relative matters: an absolute floor at a multiple of the *mean* variance destroys nearly singular contrasts, and $\Sigma = \text{diag}(10^{-8}, 10^{-8}, 1)$ makes the failure vivid — it forced $\text{Var}(X_1 - X_2)$ from 2×10^{-8} up to 6.7×10^{-4} , turning a near-certain head-to-head into a coin flip (choice error 0.48) while the projected covariance residual sat one hundred times below the warning threshold. The Gram is exactly $(1 - \frac{2}{n})I + \frac{1}{n^2}\mathbf{1}\mathbf{1}^\top$, so the constrained solve is closed form in $O(n)$: substituting $d = \ell + x$ componentwise with $x \geq 0$, writing ℓ for the vector of per-runner floors, leaves the same Gram with a shifted

the only choice-relevant quantity is the variance of the difference $X_1 - X_2$, since that alone decides the race. In correlation coordinates the fit $M = (1 - \rho)I$ scores as exact. Restored to covariance coordinates it gives a difference variance of 0.5 where the truth is 1.4, wrong by nearly a factor of three, and the correlation-space objective reports no error at all.

right-hand side, and the KKT conditions make the active set a threshold set, solved by water-filling. It is the constrained minimizer, not an unconstrained solve with a floor applied afterwards, which the Gram’s off-diagonal coupling would leave suboptimal. At $n = 2$ the Gram is rank one, only $d_1 + d_2$ is identified, and the symmetric representative is returned.

6. A second candidate, a pure eigenfit at the same total rank, closed by the same diagonal solve, is computed, and whichever candidate leaves the smaller choice-relevant residual is kept, the pipeline winning ties: the greedy factor-plus-blocks allocation is the wrong shape for globally smooth covariance, and the kernel battery below measures the repair.

The constants k , the cluster count and m are fixed, not tuned per ensemble; a spectral rule for m was tested and rejected (it saturates its cap for marginal gains and does not repair the failure case below). What results is a factor-plus-blocks race priced in one seeded, reproducible call (deterministic under Gauss–Hermite nodes, seeded scrambled Sobol past the tensor budget), and the pipeline is the shipped one-call (`fit_covariance`; `race_probabilities` accepts `cov=` directly), not a research script.

The shipped call also reports its own quality and warns on three distinct failure modes: a large choice-relevant residual (the fit could not hold the matrix); a well-fitted matrix whose pricing needs a high-rank, sharp factor integral (the quadrature, not the fit, is then the binding constraint; both regimes appear in the kernel battery below); and a nearly singular contrast the fit failed to hold. The last needs its own diagnostic, because a Frobenius or maximum-entry residual normalized by the average variance does not control choice error near a nearly singular contrast: as a pairwise difference variance approaches zero, the head-to-head probability becomes infinitely sensitive to an absolute error that global norms call negligible. The report therefore carries

$$\max_{i < j} \frac{|(e_i - e_j)^\top R (e_i - e_j)|}{(e_i - e_j)^\top \Sigma (e_i - e_j)},$$

the worst pairwise difference-variance error in units of that difference variance, which is 0.49 on the diagonal example above (where the global residual is 5×10^{-4}) and 10^{-8} once the relative floor restores the fit.

Is the fit unique? The objective is nonconvex in (V, D) , so no global certificate is claimed, and the residual report exposes the value reached. Multistart dispersion is the natural diagnostic, but it has to be measured in the right coordinates, and the obvious choice is the wrong one.

Across four ensembles at $n = 300$, with eight random diagonal starts spanning two decades of assumed idiosyncratic share, every start reaches the same objective to 4×10^{-16} relative, and no random start ever beats the shipped one. On three of the four the fits agree in choices as well, to a total variation of 10^{-14} .

Block equicorrelation is the exception, and an instructive one. There the starts agree on the objective and on D to every digit, and still price races that differ by a total variation of 0.25. The cause is degeneracy rather than optimization. Centering a six-block matrix leaves a five-fold tied leading eigenvalue, so a rank-three fit selects an arbitrary three-dimensional subspace of a five-dimensional one. Every selection is exactly optimal and each implies a different race. Raising the rank to the multiplicity makes the fit exact and collapses the disagreement to 8×10^{-4} .

Two lessons follow. Disperse the priced race across starts, not the objective, which is blind to precisely the case that matters. And when dispersion does appear, the remedy is rank rather than a better optimizer. The shipped fitter warns when the chosen rank splits a tied eigenvalue of the centered covariance, naming the rank that would take the whole tied group.

The extreme case is also the oldest model in the subject. Under equicorrelation $P\Sigma P$ is a multiple of P , so its spectrum is $(n - 1)$ -fold degenerate and a great many representations are exactly optimal: any factor rank at all, closed by a matching diagonal, reproduces $P\Sigma P$ perfectly. They are therefore all choice-equivalent, and none is wrong. But they do not cost the same to price. A representation carrying three factor columns is priced by a three-dimensional quadrature, where the equivalent pure diagonal needs none, and it pays that quadrature’s error for rank the matrix never required: measured at 1.5×10^{-5} of total variation on an eight-runner field. The fit is not what degrades; the pricing of a needlessly higher-rank description of the same race is. Preferring the lowest rank among equally optimal fits would remove it, and the warning already fires on every such matrix.

Group structures degenerate the same way, and the multiplicity is not mysterious. Equal-sized groups with equal within-group correlation leave one tied leading direction per contrast among the groups, so g equal groups give multiplicity $g - 1$, with equicorrelation the case $g = n$. The default $k = 3$ is therefore exactly right for four equal groups and misaligned outside that. Measured at $n = 24$ and within-group correlation 0.7, against the same race priced through the block grammar directly:

equal groups	leading multiplicity	rank cuts the tie	TV against grammar
2	1	no	2.4×10^{-4}
3	2	no	2.8×10^{-4}
4	3	no	1.8×10^{-4}
6	5	yes	1.0×10^{-2}
8	7	yes	3.7×10^{-2}
12	11	yes	4.0×10^{-2}

The first three rows sit at the fitter’s ordinary approximation error. The last three are the rank cutting a tied group, and the cost grows with how much of the group is cut. The warning fires on all three, and on the three-group row as well, where the error is harmless: erring toward the false positive is the right direction here.

Two things bound this. It is confined to the dense path: a structure supplied through the grammar is priced as given, with no fit, no rank to choose and no ambiguity, so §4.2 and §4.3 have no exposure. And it is a symmetry phenomenon, so it announces itself in advance: exchangeability among runners is what produces the tie, and a caller who knows the group structure knows the rank to ask for, namely one less than the number of exchangeable groups.

Accuracy at $n = 2000$. The one-call fit-and-price takes four seconds (0.6 of them the fit) against thirty-six for a 10^6 -draw simulation, and agrees with it at its noise floor on the resolved bulk.

The tail is harder to check, and we are careful about what the check establishes. For the 82 percent of alternatives with zero simulation wins, agreement is unverifiable entry by entry. Worse, that set is selected by the same simulation that would have to certify it, so a fixed-set binomial bound does not apply to it.

Sample splitting does apply, and we run it as a separate experiment (its field is configured differently, which is why its zero-win fraction differs from the one above). Fix the zero-win set on one half of the draws, then count the wins an independent second half places anywhere in that set. Because the set is fixed before the second half is looked at, its count is binomial in the set’s true mass, and a Clopper–Pearson interval is valid.

On a dense $n = 2000$ correlation with three global factors and a forty-block layer, priced in a wide field where 63.6 percent of alternatives take no win in the first million paths, the model puts 1.13×10^{-4} of mass on that set. The second million paths place 111 wins in it, a 95 percent interval of $[9.13 \times 10^{-5}, 1.34 \times 10^{-4}]$. The model’s mass is inside it.

Scored on the second half alone, so that no statistic is evaluated on the draws that selected its own target, full-vector total variation is 4.4×10^{-3} against a split-half noise floor of 5.4×10^{-3} : at the referee’s own resolution. Tail exactness at small n still comes from the Botev referee, which prices individual orthant probabilities rather than counting wins.

The identified objective earns its place. Fitting $VV' + \text{diag}(D)$ to $P\Sigma P$ is the wrong objective (§5). With the identified objective and projected residuals throughout, the ensemble study below ran a raw top- k eigenfit pipeline and the identified pipeline over twenty seeds per ensemble. On thirteen of fifteen named ensembles the arms are statistically indistinguishable and none favors the raw arm. On block equicorrelation the identified arm is eight times better at the median (2.5×10^{-3} against 2.0×10^{-2}), with a worst seed of 3.2×10^{-3} against the raw arm’s 0.162: the raw eigenfit spends its rank on a near-common component, as the identification argument predicts.

Accuracy by ensemble. Accuracy depends on the generating ensemble, so it is reported per named ensemble of the `randomcov` library (pinned at commit `0d27a51`), over *twenty seeds* per ensemble at $n = 300$, each seed refereed by its own 10^6 -path simulation (the committed `run_ensembles4.py` and `run_kernel4.py` regenerate everything).

The TV columns are full-vector total variation against the empirical frequency vector, zeros included, so the referee’s own counting noise and any unresolved tail mass contribute to the reported number. That makes the statistic upward biased in expectation, by convexity, but not a realization-wise upper bound: on a single seed sampling error can cancel part of the model error. Only the per-entry column is restricted, to entries with at least 25 referee wins. TV is reported as median, ninth decile and worst over seeds; the last column is the median per-entry absolute error on those resolved entries.

ensemble	med TV	q90 TV	worst TV	med $ p - \hat{p} $
block equicorrelation	2.5×10^{-3}	3.0×10^{-3}	3.2×10^{-3}	1.4×10^{-5}
factor + sparse links	2.9×10^{-3}	6.7×10^{-3}	1.6×10^{-2}	1.9×10^{-5}
sparse-precision graphical	4.5×10^{-3}	5.6×10^{-3}	6.0×10^{-3}	1.5×10^{-5}
Wishart	1.2×10^{-2}	1.4×10^{-2}	1.5×10^{-2}	2.6×10^{-5}
AR(1)	4.6×10^{-2}	8.4×10^{-2}	1.5×10^{-1}	7.0×10^{-5}
residuals (dense-strong)	6.8×10^{-2}	8.6×10^{-2}	9.2×10^{-2}	2.5×10^{-4}

Re-running any fixed covariance seed moves its score only at the referee’s noise scale; the medians themselves are aggregates over the twenty distinct covariance seeds, whose genuine spread the q90 and worst columns report. Eight further named ensembles (LKJ, onion, vine,

elliptope walk, animals, Marchenko–Pastur and spiked spectra, hierarchical) fall between the third and fifth rows; Archakov–Hansen sits with AR(1) at 4.9×10^{-2} .

The kernel family is the failure case. A stratified battery (kernel type $\times$ length scale $\times$ budget, twenty seeds per cell, unit-square inputs) separates two distinct mechanisms that a single “kernel” row had conflated:

kernel	length scale	med TV, $m=5$	$m=12$	$m=5$, 2^{14} nodes
RBF	0.08	3.4×10^{-2}	3.1×10^{-2}	3.3×10^{-2}
RBF	0.2	2.6×10^{-2}	2.6×10^{-2}	1.2×10^{-2}
RBF	0.4	1.6×10^{-2}	1.7×10^{-2}	8.2×10^{-3}
Matérn- $\frac{3}{2}$	0.08	2.9×10^{-2}	2.5×10^{-2}	2.4×10^{-2}
Matérn- $\frac{3}{2}$	0.2	2.5×10^{-2}	2.3×10^{-2}	1.6×10^{-2}
Matérn- $\frac{3}{2}$	0.4	2.1×10^{-2}	1.7×10^{-2}	1.2×10^{-2}

At short length scale the bottleneck is *representation*. Correlation is locality, the rank needed grows with the number of correlation lengths in the domain, and accordingly extra rank helps ($3.4 \rightarrow 3.1 \times 10^{-2}$) while extra quadrature does not.

At long length scale the fit is not the problem at all. A rank-27 eigenfit holds the RBF draw at length scale 0.4 to a projected residual of 10^{-3} . But pricing a well-fitted smooth kernel means a twenty-seven-dimensional *sharp* factor integral, since the closing diagonal is nearly zero and the conditional races are nearly deterministic. There the node budget is the binding constraint: raising the residual-factor budget from $m = 5$ to $m = 12$ does little, while 2^{14} nodes halve the error and 2^{16} nodes reach 6×10^{-3} on the diagnostic draw.

Stage 6 of the pipeline is what reaches this regime at all. The greedy factor-plus-blocks allocation leaves a projected residual of 0.14 on the same draw that the matched-rank eigenfit holds to 10^{-3} , a misallocation worth up to 7×10^{-2} of total variation, and not a limit of the grammar.

The dispatch conclusion stands. Genuinely local covariance wants sequential conditioning with cheap sparse-precision draws, and Matérn is exactly the family with the SPDE route (Lindgren et al., 2011). Near-singular smooth covariance wants either a larger node budget or plain simulation. A production front door should dispatch on the structure it detects, and the shipped call’s warnings distinguish the two regimes.

7 Benchmarks

The structure-aware per-alternative comparator. The fairest rival is not GHK but this paper’s own conditioning run the literature’s way: for each alternative separately, integrate the conditional independent-race integrand over the same Gauss–Hermite factor nodes (the construction of Butler and Moffitt (1982); `mvtnorm`’s `lpRR`/`slpRR` are its Monte Carlo form). With the same nodes and the same lattice the two computations are algebraically identical, and the measurement confirms it: at $n = 200$, rank 2, the per-alternative version agrees with the shared field to total variation 3×10^{-15} and costs 601 times as much (22s against 36ms; the committed `bench.py bm`). The comparison isolates the paper’s contribution exactly: not the conditioning, which is forty years old, but the field shared across alternatives.

Against the field. All- n choice probabilities under the rank-2 factor covariance family of §4.1, against a 2^{20} -point QMC reference. Tail is the maximum absolute log-error over reference probabilities in $[10^{-4}, 10^{-3}]$; the $n = 10$ draw has none. Frequency simulation is run at wall clock matched to the lattice and at ten times that budget; “zeros” counts tail alternatives that received no draws.

method	$n = 10$		time	$n = 30$		tail
	time	TV		TV		
lattice race	2.5 ms	1.8×10^{-4}	4.7 ms	8.3×10^{-4}	0.45, 2 zeros	0.07
Genz, full vector	46 ms	1.8×10^{-4}	1.40 s	8.3×10^{-4}		0.07
frequency, matched time	—	3.8×10^{-3}	—	1.0×10^{-2}		
frequency, 10× time	—	9.2×10^{-4}	—	2.0×10^{-3}		0.21
Mendell–Elston	2.0 ms	1.8×10^{-3}	18 ms	3.2×10^{-2}		1.30
Clark-type	1.4 ms	1.1×10^{-2}	17 ms	1.6×10^{-2}		1.33

Genz and the lattice agree to the reference’s noise and to each other. Of the $300\times$ cost ratio at $n = 30$, the one-at-a-time interface supplies an unavoidable n -fold component; the remainder reflects the differing integral representations, tolerances and implementations, and we do not apportion it further.

At the paper’s headline scale the n -fold component alone is decisive. One probability at $n = 200$ costs 0.43 s at `maxpts` = 2.5×10^5 , agreeing with the lattice to four or five digits, and six digits needs multi-second budgets (the committed `bench.py genz200`). The full vector therefore costs 86 s against 26 ms for the lattice, a factor of 3×10^3 before any accuracy matching.

The Mendell–Elston and Clark-type moment recursions are the only entries whose error grows silently: its tail entries are off by $e^{1.3} \approx 3.7$ with nothing in the output to say so. The race’s tail figure is the reference’s own noise, and lattice self-convergence places its error near 10^{-8} .

Referee agreement. A replay harness maps each benchmark probability to its difference or-thant and prices it through the two field-standard R implementations. `mvtnorm`’s Genz–Bretz agrees with the lattice to within its own reported error bound in every family: 3.8×10^{-7} against a bound of 8.1×10^{-7} at $n = 10$, and 2.7×10^{-8} against 5.8×10^{-8} on a case whose smallest probability is 3×10^{-10} . Botev’s minimax tilting agrees to its own Monte Carlo noise, which in relative terms is 2.2×10^{-4} at $p \approx 3 \times 10^{-10}$: the tail claim rests on the named accuracy reference, not only on self-convergence. On the independent family an adaptive-quadrature referee reaches machine precision and the lattice matches it to 1.9×10^{-15} . An invariance battery (two-runner closed form, translation, permutation, symmetry) holds to 10^{-16} throughout, except the two-runner comparison at 7×10^{-9} , which is the lattice discretization itself.

Stress boundaries. An adversarial battery pushes each failure axis until something gives. Depth: against pure-relative adaptive quadrature, relative error is 4×10^{-13} at $p \approx 10^{-30}$ and 2×10^{-8} at $p \approx 10^{-50}$, with the first genuine breakdown near $p \approx 10^{-119}$, a laggard twenty-five standard deviations behind. Inversion: the round trip $p \rightarrow \mu \rightarrow p$ holds 10^{-9} in log even when the target contains a 10^{-10} entry or ties at the 10^{-9} level. The Gumbel base reproduces softmax to 10^{-15} at $n = 1000$; exact duplicates split exactly, and an ϵ perturbation moves the split linearly.

The one operative boundary is factor strength. When loadings make implied correlations exceed roughly 0.9, the argmax integrand loses smoothness in factor space and Gauss–Hermite converges slowly in the node count: at loading scale 4 the 25-node rule still sits at 8×10^{-3} total variation. Scrambled-Sobol factor nodes restore reference-level accuracy (4×10^{-4}) at the same call: heavy correlation wants quasi-Monte-Carlo nodes, not deeper polynomial rules. The default escalates the family automatically past sharpness 3, in both the Python and R implementations (scrambled Sobol and Halton respectively), which agree to 5×10^{-4} on the worst case; the figures above are measured with that escalation disabled, to expose the boundary itself.

Against GHK. All- n choice probabilities under a rank-2 factor covariance, against a 2^{20} -point QMC reference. TV is a bulk metric and blind to the tails, so the same tail column as above is reported: the maximum absolute log-error over reference probabilities in $[10^{-4}, 10^{-3}]$, which for probabilities this small is the log-odds error, and which the reference’s own counting noise floors near 0.1.

n	lattice race			GHK, $R = 10^3$			GHK, $R = 10^4$		
	time	TV	tail	time	TV	tail	time	TV	tail
10	2.3 ms	1.8×10^{-4}	—	1.0 ms	8.4×10^{-3}	—	8 ms	2.1×10^{-3}	—
50	7.0 ms	1.7×10^{-3}	0.11	21 ms	4.2×10^{-2}	1.10	214 ms	1.2×10^{-2}	0.29
200	25 ms	2.4×10^{-3}	0.15	436 ms	3.2×10^{-2}	1.50	4.8 s	1.2×10^{-2}	0.35

The $n = 10$ draw has no reference probability in the tail band. The lattice’s tail figures equal the reference’s own noise; GHK’s are two to ten times larger in log terms even at ten times the draws, which is the tail blindness the TV column cannot show.

Self-convergence places the lattice error near 10^{-8} . GHK error contracts as $O(R^{-1/2})$ and its measured complete-vector cost grows superquadratically, with exponent 2.15 to 2.8 across the measured range. The lattice is deterministic under Gauss–Hermite nodes and seeded-reproducible under scrambled Sobol, supplies smooth matrix-free derivatives free of simulation noise, and prices the tails. Simulated likelihoods can of course be differentiated, but they inherit the noise.

Our claim is bounded as follows. For complete-vector pricing and inversion under the factor, block, nested and tree covariance grammars, the shared-field construction removes the per-alternative repetition intrinsic to standard GHK implementations, and dominates the per-alternative GHK implementation tested here in both accuracy and cost. GHK implementations with antithetics, quasi-random sequences or reordering heuristics would move the constants, not the n -fold shape. Estimation pipelines built around GHK are a larger object than this loop, and the estimation paragraphs below report where simulation remains competitive at face value. GHK also retains general rectangle probabilities and locality-structured covariance, and Botev (2017) remains the reference for single extreme tail probabilities.

Share inversion under sampling noise. The first estimation exercise is deliberately the saturated case. Simulate $T = 50,000$ choices from one fixed menu of $n = 30$ alternatives under a known rank-2 factor covariance and estimate the mean-zero utilities. With the menu fixed the model is saturated over the simplex interior, so the analytic-gradient maximizer is the inversion of the observed shares and the exercise measures how faithfully each method performs that inversion under multinomial noise.

Each count is smoothed by one half before any arm sees it, which is the posterior-mean share vector under the multinomial Jeffreys prior, $\mathbb{E}[p_i | c] = (c_i + \frac{1}{2}) / (T + n/2)$. (It is not the Jeffreys MAP: the Dirichlet($\frac{1}{2}$) posterior has exponents $c_i - \frac{1}{2}$, whose mode leaves empty cells on the boundary. Read as a penalized likelihood in simplex coordinates, add-half is the Dirichlet($\frac{3}{2}$) mode.) In the saturated model the estimator is the race inverse of that smoothed vector, and the simulated arms maximize the same add-half-smoothed criterion, so the comparison stays like for like.

This is not cosmetic. The unpenalized saturated likelihood has *no* finite maximizer when any count is zero, which Theorem 1 states and this design would otherwise walk into: the smallest alternative here has $p = 2.5 \times 10^{-5}$, so its expected count is 1.27 and it is empty with probability $e^{-1.27} = 0.28$, so about eleven of forty replications should see that cell empty. Thirteen of the forty do, and in this design it is the only cell that ever empties: every empty-cell replication is that alternative. On those thirteen an unpenalized optimizer returns a tolerance-dependent point on a boundary sequence rather than a maximizer. Forty replications, with the arms paired by construction since every method sees the same draw:

estimator	RMSE (median)	RMSE (mean $\pm$ s.e.)	median fit time
analytic-gradient smoothed inversion	0.0168	0.0168 ± 0.0006	2.4 s
MSL, $R = 100$	0.0286	0.0280 ± 0.0006	3.5 s
MSL, $R = 1000$	0.0184	0.0185 ± 0.0005	39.9 s

The exact path is both faster and more accurate than the $R = 100$ simulator here, and beats the $R = 1000$ accuracy in a sixteenth of its time. Because the arms share every draw, the paired differences are the sharper statement: against $R = 100$ the exact arm is better by 0.0112 ± 0.0006 , nearly nineteen standard errors, winning 39 of 40 replications; against $R = 1000$ by 0.0017 ± 0.0002 , seven standard errors, winning 33 of 40.

The $R = 1000$ margin is small in absolute terms, which is the honest characterization: enough simulation draws do approach the exact answer, at forty times the cost per fit, and the surviving paired gap is consistent with finite- R simulation bias rather than multinomial sampling noise, though pairing alone cannot rule out another systematic implementation effect. The timing column is from an unloaded machine at the original eight replications; the accuracy columns are from all forty and are insensitive to load.

Covariate estimation. The overidentified case behaves differently, and we report it as we found it. Draw observation-specific design matrices X_t ($n = 10$ alternatives, $d = 3$ covariates, $T = 800$ observations), utilities $\mu_t = X_t \beta$, one choice per observation, covariance known, and estimate β by individual-level maximum likelihood: the exact path uses lattice probabilities with the analytic score (one Jacobian row per observation, a single field pass); the simulated path uses this paper’s own Rust GHK with common random numbers and finite-difference gradients. Eight replications, means with standard errors:

estimator	RMSE($\hat{\beta}$)	median fit time
analytic-gradient MLE	0.0363 ± 0.0066	41.3 s
MSL, $R = 100$	0.0380 ± 0.0073	5.4 s
MSL, $R = 1000$	0.0367 ± 0.0067	39.6 s

All three estimators reach the statistical floor. With three parameters and eight hundred observations the simulation error averages out, and a well-implemented simulator at $R = 100$ is the fastest of the three at this size. The exact path’s time is dominated by per-observation Python overhead: the compiled kernel accounts for under a second of the 41, so the comparison at $n = 10$ measures plumbing, not method.

What the exact likelihood buys in estimation is not this row of the table. It is freedom from the choice of R , gradients without simulation noise, the cost law as n grows, and diagnostics that simulation smooths over.

That last point is not hypothetical. On the classic Fishing data, the unrestricted-covariance MNP likelihood evaluated exactly is boundary-seeking, still rising at loading norms near 4000: a ridge that the finite- R simulated objective had obscured. The behavior is reproducible from the committed estimator (`winning.mnprobit`), which detects the ridge and reports it as a `boundary_` diagnostic rather than returning the interior point simulation would have manufactured.

For interpretability of that norm: the parameterization is rank-2 loadings with the reference alternative’s rows fixed at zero and unit idiosyncratic variances, so global scale is pinned by the idiosyncratic normalization and the growing norm is a genuine choice-relevant direction, not the scale gauge. A profile likelihood along that normalized direction is the diagnostic a full treatment would add.

The Genz–Bretz score. Our quadrature comparator used `pmvnorm`, which exposes no score, with finite-difference gradients (31 vector evaluations per step at $n = 30$, roughly eight times the entire exact-gradient share-inversion fit). The `mvtnorm` manual (v1.4-2) also documents `slpmvnorm` and, for the reduced-rank covariance $BB' + \text{diag}(D)$ of this paper’s benchmarks, `lpRR` and `slpRR`: simulated log-likelihoods and score functions that integrate over the K factor dimensions. That interface is itself the conditioning construction of §4.1 with Monte Carlo in place of quadrature and no shared field across alternatives, which makes it the natural structure-aware comparator for a fuller estimation study.

GHK protocol. The GHK figures throughout use this paper’s own Rust implementation: per-alternative sequential conditioning on the reference-differenced covariance with a fresh Cholesky factorization per alternative, pseudorandom draws (xoshiro family) with a per-alternative seed offset so that parameter evaluations share common random numbers, no antithetics, no quasi-random sequence, no variable-reordering heuristic, parallelized across alternatives. Cost figures are complete-vector; the large- n columns of the introduction are extrapolations of the measured law and are displayed as such.

Scale. Ten million contestants price in 245 seconds under the block grammar (ten thousand clusters, 257-point lattice, the streaming field pass; the committed `bench.py tenmillion`), and the Rust classic-calibration kernel runs 232 times the speed of the pure-Python implementation. Throughput at this size is a systems statement; the accuracy evidence lives at the scales the referees above can reach.

Parity. One vector file embeds the inputs and outputs of 22 scenarios spanning the principal implemented grammars, boundary cases and numerical identities used above. The four lan-

guage implementations of §9 replay them: nineteen to within 10^{-7} (most to 10^{-10}), and the three fit- or optimizer-mediated scenarios to between 5×10^{-4} and 5×10^{-3} .

8 Summary

The pieces of this paper are individually old. The one-dimensional representation of the choice event is in Domencich and McFadden (1975), who judged it computationally intractable for correlated probit; the convex duality of calibration is in Li (2018); the tie-density derivatives and the surplus function’s Laplacian structure are in Müller et al. (2022); a Hessian–vector product at the cost of a gradient is Pearlmutter (1994). The innovation is their organization: a single shared survival field renders probabilities for all n alternatives at once, supplies the own-coordinate slopes that precondition the million-alternative calibrations reported here, and applies the dense tie-density Jacobian — a weighted graph Laplacian that is a probability flux through the boundary between winning regions — in linear work per lattice point. Within the covariance grammar that organization prices, differentiates and inverts contests at a scale on which the representation had previously stopped being an algorithm.

9 Availability

`pip install winning` is pure Python; `winning[fast]` adds the Rust kernels as one abi3 wheel per platform. A base-R package mirrors the full interface, and a zero-dependency browser port drives a live demonstration at winning.microprediction.org.

References

- S. D. Ahipasaoglu, X. Li, K. Natarajan. A convex optimization approach for computing correlated choice probabilities with many alternatives. *IEEE Transactions on Automatic Control* 64:190–205, 2019.
- J. H. Albert, S. Chib. Bayesian analysis of binary and polychotomous response data. *JASA* 88:669–679, 1993.
- S. Berry. Estimating discrete-choice models of product differentiation. *RAND Journal of Economics* 25(2):242–262, 1994.
- J. S. Butler and R. Moffitt. A computationally efficient quadrature procedure for the one-factor multinomial probit model. *Econometrica* 50(3):761–764, 1982.
- W. Benter. Computer based horse race handicapping and wagering systems: a report. In D. B. Hausch, V. S. Y. Lo, W. T. Ziemba (eds.), *Efficiency of Racetrack Betting Markets*, Academic Press, 1994, 183–198.
- Z. I. Botev. The normal law under linear restrictions: simulation and estimation via minimax tilting. *Journal of the Royal Statistical Society B* 79(1):125–148, 2017.
- A. Börsch-Supan, V. Hajivassiliou. Smooth unbiased multivariate probability simulators for maximum likelihood estimation of limited dependent variable models. *Journal of Econometrics* 58:347–368, 1993.

- M. Batram, D. Bauer. On consistency of the MACML approach to discrete choice modelling. *Journal of Choice Modelling* 30:1–16, 2019.
- D. Bauer. *Rprobit: R package for the estimation of multinomial probit models*, 2024. github.com/dbauer72/Rprobit.
- C. R. Bhat. The maximum approximate composite marginal likelihood (MACML) estimation of multinomial probit-based unordered response choice models. *Transportation Research Part B* 45(7):923–939, 2011.
- L. Cappellari, S. P. Jenkins. Multivariate probit regression using simulated maximum likelihood. *Stata Journal* 3(3):278–294, 2003.
- K. Chiong, A. Galichon, M. Shum. Duality in dynamic discrete-choice models. *Quantitative Economics* 7(1):83–115, 2016.
- K. Chellel. The gambler who cracked the horse-racing code. *Bloomberg Businessweek*, May 3, 2018.
- C. E. Clark. The greatest of a finite set of random variables. *Operations Research* 9:145–162, 1961.
- P. Cotton. Probit versus logit: a survey of empirical accuracy. SSRN working paper, 2026.
- Y. Croissant. Estimation of random utility models in R: the mlogit package. *Journal of Statistical Software* 95(11):1–41, 2020.
- P. Cotton. Inferring relative ability from winning probability in multi-entrant contests. *SIAM Journal on Financial Mathematics* 12(1):295–317, 2021.
- J. M. Danskin. The theory of max-min, with applications. *SIAM Journal on Applied Mathematics* 14(4):641–664, 1966.
- M. Fosgerau, D. McFadden, M. Bierlaire. Choice probability generating functions. *Journal of Choice Modelling* 8:1–18, 2013.
- A. Galichon, B. Salanié. Cupid’s invisible hand: social surplus and identification in matching models. *Review of Economic Studies* 89(5):2600–2629, 2022.
- R. Gates. A Mata Geweke–Hajivassiliou–Keane multivariate normal simulator. *Stata Journal* 6(2):190–213, 2006.
- D. A. Harville. Assigning probabilities to the outcomes of multi-entry competitions. *Journal of the American Statistical Association* 68(342):312–316, 1973.
- T. A. Domencich, D. McFadden. *Urban Travel Demand: A Behavioural Analysis*. North-Holland, Amsterdam, 1975.
- A. Genz. Numerical computation of multivariate normal probabilities. *Journal of Computational and Graphical Statistics* 1:141–149, 1992.
- J. Geweke. Bayesian inference in econometric models using Monte Carlo integration. *Econometrica* 57:1317–1339, 1989.
- V. Hajivassiliou, D. McFadden, P. Ruud. Simulation of multivariate normal rectangle probabilities and their derivatives. *Journal of Econometrics* 72:85–134, 1996.
- K. Imai, D. A. van Dyk. MNP: R package for fitting the multinomial probit model. *Journal of Statistical Software* 14(3):1–32, 2005.

- J. D. Kalbfleisch, R. L. Prentice. *The Statistical Analysis of Failure Time Data*. 2nd ed., Wiley, 2002.
- M. P. Keane. A computationally practical simulation estimator for panel data. *Econometrica* 62:95–116, 1994.
- J. Kitagawa, Q. Mérigot, B. Thibert. Convergence of a Newton algorithm for semi-discrete optimal transport. *Journal of the European Mathematical Society* 21(9):2603–2651, 2019.
- S. Lerman, C. Manski. On the use of simulated frequencies to approximate choice probabilities. In *Structural Analysis of Discrete Data*, MIT Press, 1981.
- B. Lévy, R. Mohayaee, S. von Hausegger. A fast semidiscrete optimal transport algorithm for a unique reconstruction of the early Universe. *Monthly Notices of the Royal Astronomical Society* 506(1):1165–1185, 2021.
- L. Li. A general method for demand inversion. arXiv:1802.04444 (2018).
- F. Lindgren, H. Rue, J. Lindström. An explicit link between Gaussian fields and Gaussian Markov random fields. *Journal of the Royal Statistical Society B* 73(4):423–498, 2011.
- R. Loaiza-Maya, D. Nibbering. Scalable Bayesian estimation in the multinomial probit model. *Journal of Business & Economic Statistics* 40(4):1678–1690, 2022.
- M. López de Prado. Building diversified portfolios that outperform out of sample. *Journal of Portfolio Management* 42(4):59–69, 2016.
- R. D. Luce. *Individual Choice Behavior: A Theoretical Analysis*. Wiley, 1959.
- D. McFadden. Econometric models of probabilistic choice. In C. Manski, D. McFadden, eds., *Structural Analysis of Discrete Data*, MIT Press, 1981.
- R. McCulloch, P. Rossi. An exact likelihood analysis of the multinomial probit model. *Journal of Econometrics* 64:207–240, 1994.
- N. Mendell, R. Elston. Multifactorial qualitative traits: genetic analysis and prediction of recurrence risks. *Biometrics* 30:41–57, 1974.
- V. K. Mishra, K. Natarajan, H. Tao, C.-P. Teo. Choice prediction with semidefinite optimization when utilities are correlated. *IEEE Transactions on Automatic Control* 57:2450–2463, 2012.
- F. Mosteller. Remarks on the method of paired comparisons: I. The least squares solution assuming equal standard deviations and equal correlations. *Psychometrika* 16:3–9, 1951.
- D. Müller, Y. Nesterov, V. Shikhman. Discrete choice prox-functions on the simplex. *Mathematics of Operations Research* 47(1):485–507, 2022.
- B. A. Pearlmutter. Fast exact multiplication by the Hessian. *Neural Computation* 6(1):147–160, 1994.
- A. R. Solow. A method for approximating multivariate normal orthant probabilities. *Journal of Statistical Computation and Simulation* 37:225–229, 1990.
- S. Stern. A method for smoothing simulated moments of discrete probabilities in multinomial probit models. *Econometrica* 60:943–952, 1992.
- StataCorp. *Stata Choice Models Reference Manual: cmmprobit, asmprobit*. Stata Press, College Station, 2023.
- L. L. Thurstone. A law of comparative judgment. *Psychological Review* 34:273–286, 1927.
- K. Train. *Discrete Choice Methods with Simulation*. 2nd ed., Cambridge, 2009.

A Extrapolated GHK cost brackets

The lattice column below is measured; the GHK column is an extrapolation of the two measured exponents (2.15 and 2.8) far beyond any regime in which GHK has been run, and cache effects, parallelism, Cholesky reuse or a change of regime would move it.

n	lattice, measured	GHK at 10^4 draws, cost-law bracket	multiplier
10^4	0.18 s	20 hours – 3.7 days	$4 \times 10^5 - 2 \times 10^6$
10^5	2.7 s	0.3 – 6.5 years	$4 \times 10^6 - 8 \times 10^7$
10^6	29 s	46 – 4,100 years	$5 \times 10^7 - 4 \times 10^9$

Nobody prices a million-alternative field with GHK; the brackets say what the measured law implies about why. The ten-million forward-field benchmark of §7, which is actually run, is the meaningful systems demonstration.