A Higher Categorical Theory of Polynomial-Branch Recursive Boosting with Maximum-Entropy Error Selection

Anonymous

November 10, 2026

Abstract

We construct a typed theory of recursive boosting whose candidate branches grow polynomially in the number of gates at fixed interaction degree. Finitary polynomial functors generate syntax in an infinity-topos; evaluation, graded substitution, and descent relate higher presentations to finite empirical optimization. Maximum-entropy error selection gives Gibbs weights and a convex tilted risk. Conditional-gradient fitting has explicit oracle-error bounds, a computable primal-gap certificate, and linear growth of the active branch count. An auxiliary-variable identity supplies unbiased stochastic gradients for the same entropic objective, enabling controlled comparisons with single-example and minibatch SGD. We supply rational-arithmetic code, numerical implementations, synthetic examples, and six undergraduate learning comparisons. The polynomial degree, scalar truncation, numerical precision, and limits of unrestricted branch search are stated explicitly.

Keywords: recursive boosting; polynomial functors; infinity-topos; entropy; conditional gradient; stochastic gradient descent.

1 Problem and mathematical contract

Recursive boosting repeatedly fits errors using predictors that can themselves contain gated subpredictors and additive ensembles. Without a growth discipline, this description permits exponentially many syntactic branches. Without an oracle contract, it can also conceal an intractable search inside a purportedly efficient fitting procedure. We give separate, explicit bounds for candidate generation, active model size, empirical optimization, and representation transport.

Fix a nonempty sample $D = ((x_i, y_i))_{i=1}^n$, b bounded gates $g_j : X \to [0,1]$, and H bounded leaves $h_r : X \to [-1,1]$. A branch is a leaf multiplied by a finite sequence of gates. Addition combines branches; scalar multiplication supplies signed coefficients. A fixed integer k bounds the total number of gate multiplications on any branch. The cost of a branch can increase with its degree. The effective specialization uses finite encodings of gate and leaf values on the sample.

The main optimization problem is a convex relaxation with a variation-norm radius, rather than the nonconvex problem of choosing the best predictor with exactly B branches. Nevertheless its optimizer is approached by an algorithm whose iterate after T updates contains at most T branch coordinates. Thus the relaxation is accompanied by a concrete sparse construction and an error certificate.

Contributions. The contribution is the combination of a graded higher syntax, a canonical polynomial dictionary, maximum-entropy error selection, an effective fitting algorithm, and comparisons against SGD on an identical objective. The conditional-gradient rate is a specialization of established convex optimization, not a new general rate. Entropic risk is a known tilted objective. We prove the particular reduction, cost accounting, transport statements, and certificates used here.

Prior work. Schapire's weak-to-strong learning construction [1] and Freund–Schapire boosting [2] establish the classical background. Functional optimization is central to Friedman's gradient boosting [3]. Jaggi [4] analyzes conditional gradients, approximate linear oracles, and primal gaps. AdaNet [5] studies adaptive neural architecture selection. Our grammar is narrower than an arbitrary neural architecture search space and therefore admits direct finite enumeration.

Polynomial functors and free monads are developed in ordinary category theory by Gambino–Kock [6], and for infinity-groupoids by Gepner–Haugseng–Kock [7]. We use the elementary finitary tree construction in an infinity-topos, whose descent and truncation structure follows the framework of [8]. Maximum-entropy weighting is expressed as tilted empirical risk, as in [9]. These sources provide the context; the proofs below spell out the finite algorithmic consequences.

1.1 Notation and standing hypotheses

For sample vectors $u, v \in \mathbb{R}^n$, put

$$\langle u, v\rangle_n = \frac{1}{n}\sum_{i=1}^n u_i v_i, \qquad \|u\|_n^2 = \langle u, u\rangle_n.$$

For vector outputs in $\mathbb{R}^C$, replace the product by the Euclidean dot product in each row. Gradients are taken in this empirical inner product. In particular, for $F(z) = n^{-1}\sum_i \ell(y_i, z_i)$, the gradient has entries $\ell_i'(z_i)$, without an additional factor $1/n$.

All gate and leaf families are fixed during one optimization run. A separate dictionary expansion changes the comparison class and must be recorded as such. In the effective results, k is fixed, the sample and dictionary are finite, and arithmetic is either exact rational arithmetic or computable approximation with explicit error allowances. Analytic results use a convex twice differentiable loss on the bounded prediction domain; extension to a continuously differentiable smooth loss follows by the same smoothness inequality.

The infinity-topos semantics does not turn arbitrary real objects into computable data. It supplies a language for presentations, symmetries, and change of context. The finite algorithm is obtained by an explicitly selected computational interpretation.

2 Canonical recursive branches

Definition 2.1 (Recursive atom)**.** For $\alpha \in \mathbb{N}^b$, write $|\alpha| = \sum_j \alpha_j$ and

$$u_{\alpha,r}(x) = h_r(x)\prod_{j=1}^{b} g_j(x)^{\alpha_j}.$$

Let $c_{\alpha,r} \geq 1$ be a computable positive cost, and let $a_{\alpha,r} = u_{\alpha,r}/c_{\alpha,r}$. The unsigned dictionary $\mathcal{A}_k$ consists of all labelled pairs (α, r) with $|\alpha| \leq k$, interpreted by $a_{\alpha,r}$. Distinct labels may evaluate to the same predictor.

Repeated gates are allowed. Sorting a gate word produces its exponent vector because scalar gate multiplication is commutative. Repetition is not removed for soft gates; g^2 need not equal g. Boolean gates do admit the additional idempotence reduction. The implementation keeps repeated-gate labels, so its dictionary sizes agree with the formula below on both Boolean and soft inputs.

Theorem 2.2 (Polynomial candidate count)**.** *The number of unsigned labelled atoms is*

$$M = H\sum_{d=0}^{k}\binom{b+d-1}{d} = H\binom{b+k}{k}.$$

For fixed k, $M = O(H(b+1)^k)$. For Boolean gates with idempotence imposed, a possibly smaller dictionary has at most $H\sum_{d=0}^{\min(k,b)}\binom{b}{d}$ labels.

Proof. A degree-d exponent vector is a multiset of d elements from a b-element set, counted by stars and bars. Summing the counts uses the hockey-stick identity. Equivalently append a slack coordinate $\alpha_{b+1} = k - |\alpha|$, and count nonnegative solutions of $\sum_{j=1}^{b+1}\alpha_j = k$. For $b = 0$, the empty product gives $M = H$, in agreement with $\binom{k}{k} = 1$. Boolean idempotence restricts each exponent to 0 or 1. The polynomial estimate follows directly for fixed k. □

The recursion underlying this dictionary is $u \mapsto g_j u$. A canonical implementation appends only a gate index no smaller than the preceding one. Every multiset appears once, and deleting the last gate gives its unique cached prefix. This is an effective recursive procedure, rather than an instruction to search over arbitrary real-valued functions.

Definition 2.3 (Feasible empirical predictions)**.** Let $A \in \mathbb{R}^{n\times M}$ contain the normalized sample atoms. Define

$$\mathcal{K}_{R,k} = \{A\theta : \|\theta\|_1 \leq R\} = R\,\mathrm{conv}\big(\{0\}\cup\{\pm A_j\}_{j=1}^{M}\big).$$

In raw coefficients $\beta_{\alpha,r} = \theta_{\alpha,r}/c_{\alpha,r}$, feasibility is $\sum c_{\alpha,r}|\beta_{\alpha,r}| \leq R$.

The zero vertex can be omitted when the signed set is nonempty because it is already its midpoint. Keeping it makes initialization explicit. Every feasible prediction satisfies $|z_i| \leq R$ and $\|z\|_n \leq R$. For multiclass fitting the atoms are $A_j e_c$, the coefficient matrix has total entrywise ℓ^1 norm at most R, and each row has Euclidean norm at most R.

Proposition 2.4 (Existence and increasing expressivity)**.** *A continuous empirical objective attains a minimum on $\mathcal{K}_{R,k}$. If the cost assigned to an existing atom is unchanged when k increases, then $\mathcal{K}_{R,k} \subseteq \mathcal{K}_{R,k+1}$, and the optimal value cannot increase.*

Proof. The coefficient ball is compact in finite dimension and its linear image is compact. Apply the extreme-value theorem. Zero padding embeds the lower-degree coefficient ball in the larger one without changing its norm. □

Lemma 2.5 (Flattening)**.** *Every finite expression generated by bounded leaves, gate multiplication, addition, and scalar multiplication, with at most k gates on each leaf-to-root path, evaluates to $\sum_{|\alpha|\leq k,r}\beta_{\alpha,r}u_{\alpha,r}$. Its raw weighted variation is bounded by the sum of the weighted absolute path coefficients before like terms are combined.*

Proof. Induct on the expression. A leaf gives the empty exponent vector. A gate increments one exponent. Addition concatenates the two finite lists, and a scalar multiplies every coefficient. Distributivity gives equality of evaluations at each step. Combining equal labels only decreases the sum of absolute coefficients by the triangle inequality. Assigning each expanded path its own prescribed cost gives the claimed bound. □

The last statement does not assert that arbitrary syntactic scalar and gate operations preserve a fixed radius automatically. Feasibility is a separate certificate attached

to the flattened coefficients. In particular, multiplying two already boosted predictors is not a primitive operation of the fitting grammar: it would multiply supports and add degrees. Such a product can be recompiled only after its cost and degree have been checked.

Example 2.6 (A genuinely recursive interaction)**.** For $x_1, x_2 \in \{0,1\}$, the signed XOR target

$$y = (1-2x_1)(1-2x_2) = 1 - 2x_1 - 2x_2 + 4x_1x_2$$

is a degree-two recursive branch expansion. With $c_\alpha = 1 + |\alpha|$, its variation is $1+4+4+12 = 21$. It is feasible for $R = 24$. On the uniform four-point sample, y is orthogonal to $1, x_1, x_2$, so the best degree-one square-loss predictor is zero, with loss $1/2$. Degree two attains zero loss. This separates a representation benefit from an optimizer comparison.

3 Higher categorical syntax and semantics

3.1 A finitary polynomial signature

Let $\mathcal{X}$ be an infinity-topos. Work in a slice $\mathcal{X}_{/U}$ when a parameter context U is needed. Choose a 0-truncated ordered ring object $\mathbf{R}$, a data object $\mathbf{X}$, a gate object $\mathbf{G}$, and a leaf object $\mathbf{H}$. Their interpretations are maps

$$\mathbf{G} \times \mathbf{X} \longrightarrow [0,1]_{\mathbf{R}}, \qquad \mathbf{H} \times \mathbf{X} \longrightarrow [-1,1]_{\mathbf{R}}.$$

The labels $\mathbf{G}, \mathbf{H}$ may have higher automorphisms. A map into the scalar object still has to respect them. For analytic semantics we may take the constant discrete real ring; an effective subinterpretation uses rational labels and evaluations. No completeness property of an arbitrary internal ordered ring is used in constructing syntax.

Consider the endofunctor

$$P(Z) = 1 \amalg (\mathbf{G} \times Z) \amalg (\mathbf{R} \times Z) \amalg Z^2. \tag{1}$$

The four summands represent zero, gate multiplication, scalar multiplication, and addition. This is polynomial in the categorical sense: there is a diagram $1 \leftarrow E \rightarrow B \rightarrow 1$ whose fibers of input positions have cardinalities $0, 1, 1, 2$, and $P = \Sigma_B \Pi_{E/B} \Delta_E$. The coefficient objects are labels for these arities. This use of "polynomial" concerns the signature; Theorem 2.2 is a separate numerical polynomial-growth statement.

Theorem 3.1 (Free recursive syntax)**.** *The finitary signature* (1) *admits a free-algebra monad* T_P *on* $\mathcal{X}$*. Its value on* V *is the colimit of*

$$T_0(V) = V, \qquad T_{m+1}(V) = V \amalg P(T_m(V)).$$

Ground boosting syntax is $\mathsf{T}_P(\mathbf{H})$*. Every interpretation of the four operations on an object* W*, together with* $V \rightarrow W$*, extends to an algebra map* $\mathsf{T}_P(V) \rightarrow W$*, with a contractible space of compatible choices.*

Proof. Finite products commute with filtered colimits in an infinity-topos. Coproducts and multiplication by a fixed object preserve colimits. Consequently P preserves the sequential colimit of this chain. The induced map $V \amalg P(\operatorname{colim}_m T_m(V)) \rightarrow \operatorname{colim}_m T_m(V)$ is an equivalence by shifting the chain. A map to a proposed algebra is built recursively on T_m, and compatibility fixes its extension at every stage. Taking the limit of these compatible mapping spaces yields the initial-algebra universal property. Applying this construction functorially in V gives the unit by inclusion of variables and the multiplication by substitution. Both monad identities identify the same free-algebra extension, so their coherent compatibility follows from its contractible uniqueness. □

This is a finite-tree construction with potentially higher spaces of labels, not a declaration that all syntax is a set. It specializes to familiar W-type syntax when the labels and ambient semantics are 0-truncated.

3.2 Evaluation and the scalar truncation

The object $\mathbf{R}^{\mathbf{X}}$ carries the operations in (1): pointwise zero, pointwise addition and scaling, and pointwise multiplication by a gate. The leaf interpretation therefore induces

$$\mathrm{ev} : \mathsf{T}_P(\mathbf{H}) \longrightarrow \mathbf{R}^{\mathbf{X}}.$$

For sample points $x_1, \ldots, x_n$ it factors into an empirical interpretation

$$\mathrm{ev}_D : \mathsf{T}_P(\mathbf{H}) \longrightarrow \mathbf{R}^n.$$

Substitution is respected by evaluation because both ways of interpreting a substituted tree are algebra maps agreeing on generators. At the level of coherent maps this is the universal property of the free monad, rather than a new ad hoc interchange axiom.

Proposition 3.2 (Scalar invariance)**.** *The object* $\mathbf{R}^{\mathbf{X}}$ *is* 0*-truncated. Every path between presentations has identical scalar predictions; every higher homotopy between such paths has the same scalar image. The evaluation map factors uniquely through* $\tau_{\leq 0}\mathsf{T}_P(\mathbf{H})$*.*

Proof. If Z is 0-truncated then, for every V, the mapping space $\mathrm{Map}(V, Z^{\mathbf{X}}) \simeq \mathrm{Map}(V \times \mathbf{X}, Z)$ is 0-truncated. Hence $Z^{\mathbf{X}}$ is 0-truncated. The universal property of truncation gives the factorization. A path in a discrete target is an equality, and its higher path spaces are contractible when nonempty. □

Example 3.3 (Higher presentations invisible to the loss)**.** Take $\mathcal{X} = \mathcal{S}$, $\mathbf{H} = BC_2$, and interpret every leaf by the constant function 1. The variable-leaf summand of $\mathsf{T}_P(\mathbf{H})$ retains the nontrivial order-two automorphism. Its scalar evaluation is constant. Thus the presentation space contains information that ordinary empirical risk cannot distinguish. Higher structure records symmetries and compatible implementations; scalar optimization takes place after truncation.

3.3 Resource grades and typed substitution

Attach to a finite presentation t a branch count $B(t)$ and a maximum gate depth $d(t)$. A variable leaf has grade $(1,0)$; zero has $(0,0)$. Scaling preserves the grade, gating sends (B,d) to $(B,d+1)$, and addition sends $(B_1,d_1),(B_2,d_2)$ to $(B_1+B_2,\max(d_1,d_2))$. The count belongs to a presentation, rather than to an unlabelled numerical function.

Proposition 3.4 (Substitution bound)**.** *Suppose $B(t)\le B$, $d(t)\le k$, and every replacement of a variable leaf has branch count at most C and depth at most l. Then*

$$B(t[\mathbf{u}])\le BC,\qquad d(t[\mathbf{u}])\le k+l.$$

These bounds are stable under iterated substitution, with coherent grade multiplication $(B,k)\odot(C,l)=(BC,k+l)$.

Proof. Each of at most B variable leaves contributes at most C replacement leaves. Every resulting path is a path in t followed by a path in one replacement. Its gate length is at most $k+l$. Associativity of multiplication and addition of natural numbers agrees with associativity of tree substitution. Structural induction gives compatibility for addition and each unary constructor. □

This proposition explains precisely where exponential expansion can occur: substituting a two-branch expression into every leaf at every level gives the upper bound 2^d. Polynomial growth requires a restriction on this process. The fitting algorithm instead adds one canonical atom per round, preserving $d\le k$ and increasing B by at most one.

Algebraic refactorings can change node counts and even branch counts through cancellation. We therefore retain an explicit resource witness. A numerical equality alone is not evidence that an arbitrary inverse rewrite respects a budget. For transport inside a fixed budget, morphisms must include a budget-preservation witness. Ordered gate words are mapped to canonical multisets by the scalar interpreter; the free tree monad itself does not silently impose commutativity of gate labels.

3.4 A sound resource type system

For the concrete cost $c_\alpha=1+|\alpha|$, a more informative judgment is

$$\Gamma\vdash t:\mathrm{Pred}(B,k,U,V).$$

Here B bounds expanded leaves, k bounds gate depth, U bounds the sum of absolute expanded path coefficients, and V bounds their degree-weighted sum. A leaf has type $(1,0,1,1)$ and zero has type $(0,0,0,0)$. Addition combines types by

$$\begin{aligned}&(B,k,U,V)+(B',k',U',V')\\&=(B+B',\max(k,k'),U+U',V+V').\end{aligned}$$

Scaling by λ multiplies U,V by $|\lambda|$. A gate changes (B,k,U,V) to $(B,k+1,U,V+U)$. These rules are finite, syntax directed, and computable on rational coefficient labels. They permit weakening to any larger bounds.

Theorem 3.5 (Type and compilation soundness)**.** *If $\Gamma\vdash t:\mathrm{Pred}(B,k,U,V)$, flattening and canonicalizing its gate words produces a coefficient vector over $\mathcal{A}_k$ with at most B terms and weighted variation at most V. Its predictions equal the recursive evaluation. Thus $V\le R$ certifies membership in $\mathcal{K}_{R,k}$.*

Proof. Track the expanded paths before cancellation. A leaf contributes coefficient mass one and weighted mass one. A scalar multiplies both masses by its absolute value. Addition sums masses. Gating increases the degree of each path by one, so its weighted mass increases by exactly its unweighted mass. This proves the rules by induction. Sorting each word preserves its length, scalar product, and cost. Combining like words uses the triangle inequality and cannot increase the weighted mass. The result is precisely the coefficient condition in Definition 2.3; Lemma 2.5 proves equality of predictions. □

For finite discrete gate labels, this compilation factors through the free module with basis $h_r g^\alpha$ of total gate degree at most k. Equivalently, it is the degree-$\le k$ part of $\mathbb{R}[g_1,\dots,g_b]\otimes\mathbb{R}^H$, interpreted on the sample. The kernel of empirical evaluation includes any sample-specific identities, such as coincident columns. The canonical module does not quotient by unknown sample identities during candidate counting.

Convex coefficient updates are interpreted in the finite-distribution algebra of empirical predictions. Gate multiplication is linear and therefore preserves every finite convex combination. This distributivity explains why nested gated ensembles can be flattened before fitting. The higher tree presentation still remembers how the distributivity was implemented; the scalar linear interpreter forgets that presentation data.

3.5 Descent and change of context

For any object of $\mathcal{X}$, its spaces of sections satisfy descent. In particular, a presentation assembled over a cover is meaningful only when its restrictions come with coherent equivalences on overlaps. Pairwise agreement without the higher coherence data is insufficient for an infinity-groupoid of labels.

Theorem 3.6 (Transport of finite recursive syntax)**.** *Let $f^*:\mathcal{X}\to\mathcal{Y}$ be the inverse-image functor of a geometric morphism. There is a natural equivalence*

$$f^*\mathsf{T}_P(V)\simeq\mathsf{T}_{f^*P}(f^*V),$$

compatible with the monad unit, substitution, and finite presentation grades. Finite sample evaluation commutes with transport through the canonical comparison of prediction objects.

$$\begin{array}{ccc} f^*\mathsf{T}_P(\mathbf{H}) & \xrightarrow{\simeq} & \mathsf{T}_{f^*P}(f^*\mathbf{H}) \\ {\scriptstyle f^*\,\mathrm{ev}_D}\downarrow & & \downarrow{\scriptstyle \mathrm{ev}_{f^*D}} \\ f^*(\mathbf{R}^n) & \xrightarrow[\simeq]{} & (f^*\mathbf{R})^n \end{array}$$

Figure 1: Finite empirical interpretation is compatible with change of context. No preservation of arbitrary exponentials is assumed.

Proof. The functor f^* preserves finite limits and all colimits. It therefore preserves the finite products and coproducts defining P, the chain defining $\mathsf{T}_P(V)$, and its colimit. Induction identifies the two chains and yields the displayed equivalence. Unit and substitution commute by their free-algebra characterizations. Grade computation uses finite tree recursion, so its witness transports as well. Finally $f^*(\mathbf{R}^n) \simeq (f^*\mathbf{R})^n$ because the exponent here is a fixed finite set. □

For clarity, a geometric inverse image need not preserve an arbitrary exponential. For a general data object there is a comparison $f^*(\mathbf{R}^{\mathbf{X}}) \to (f^*\mathbf{R})^{f^*\mathbf{X}}$, not an asserted equivalence. The theorem uses this map for general prediction objects and uses a genuine equivalence only for the finite sample target.

The space of presentations over a fixed prediction z is the homotopy fiber of ev_D at z. This fiber may be higher even though z is discrete. In a sheaf model these fibers glue with their equivalences. In a finite executable interpretation only the selected labels, coefficient encodings, and sample values enter the optimizer.

Remark 3.7 (Optimization is a correspondence at ties)**.** The exact oracle is naturally the set of minimizing labelled signed atoms. A deterministic enumeration chooses one member but is not invariant under every relabelling. For two tied atoms exchanged by a symmetry, no equivariant single-atom selector exists unless the symmetry fixes an atom. Keeping the whole minimizing set, or sampling uniformly from it, preserves that finite symmetry. We do not claim that arbitrary deterministic tie-breaking is a natural transformation of higher contexts.

4 Maximum-entropy selection of fitting errors

4.1 The error-selection principle

For the present predictor z, define nonnegative example losses $e_i = \ell(y_i, z_i)$. We maximize the entropy of the example-selection distribution while rewarding error coverage. With $u_i = 1/n$ and temperature $\tau > 0$, solve

$$\max_{p \in \Delta_n} \left\{ \sum_i p_i e_i - \tau\,\mathrm{KL}(p\|u) \right\}. \tag{2}$$

Since $\mathrm{KL}(p\|u) = \log n - \mathsf{H}(p)$, this objective is mean selected error plus τ times Shannon entropy, up to a constant. Pure entropy maximization would select uniformly and would not prioritize errors. The error term supplies that priority, while the entropy term prevents a hard collapse onto a single worst example.

Theorem 4.1 (Gibbs error selector)**.** *The unique maximizer of* (2) *and its value are*

$$p_i^*(z) = \frac{\exp(e_i/\tau)}{\sum_j \exp(e_j/\tau)}, \tag{3}$$

$$F_\tau(z) = \tau \log\left(\frac{1}{n}\sum_i \exp(e_i/\tau)\right). \tag{4}$$

Furthermore, for any $p \in \Delta_n$,

$$F_\tau - \left(\sum_i p_i e_i - \tau\,\mathrm{KL}(p\|u)\right) = \tau\,\mathrm{KL}(p\|p^*).$$

Proof. Substitute $\log p_i^* = e_i/\tau - \log\sum_j \exp(e_j/\tau)$ into the right side and expand. All terms cancel to give the identity. Nonnegativity of relative entropy gives optimality, and equality occurs only at $p = p^*$. The formula includes boundary distributions using $0 \log 0 = 0$. □

There is also a literal constrained maximum-entropy interpretation. Put $m = \sum_i p_i^* e_i$. Among distributions with selected mean error m, p^* maximizes entropy. Indeed the first and normalization terms in the identity are fixed on that affine constraint, so any increase in entropy would contradict optimality. If all errors are equal, the maximizer is uniform.

Proposition 4.2 (Concentration controls)**.** *If $0 \le e_i \le B$, then*

$$\begin{aligned} \frac{1}{n}\sum_i e_i \le F_\tau \le \max_i e_i, \\ \max_i e_i - \tau \log n \le F_\tau, \\ n \max_i p_i^* \le \min\{n, \exp(B/\tau)\}, \\ \mathsf{H}(p^*) \ge \log n - B/\tau. \end{aligned}$$

The entropy effective count $\exp \mathsf{H}(p^*)$ is at least $n\exp(-B/\tau)$ and at most n.

Proof. Jensen's inequality gives the first lower bound; bounding every exponential by the largest gives the upper bound. Retaining just one maximal term gives the second lower bound. The Gibbs denominator is at least n, proving the concentration estimate. Finally $\mathrm{KL}(p^*\|u) = (\sum_i p_i^* e_i - F_\tau)/\tau \le B/\tau$. Subtract from $\log n$ and exponentiate. □

These bounds can be loose. The implementation records the actual entropy, effective count, and gap. Small τ emphasizes large errors and can emphasize corrupted labels as well. A claim of automatic robustness to outliers would reverse the meaning of the positive tilt used here.

Loss	L_0	G_0 on the feasible set
Square $\frac{1}{2}(z-y)^2$	1	$R+Y$
Binary logistic $\log(1+e^{-yz})$	1/4	1
Multiclass cross entropy	1/2	$\sqrt{2}$

Table 1: Constants for Theorem 4.3. Square loss assumes $|y| \le Y$; binary labels are ± 1. Multiclass scores use the Euclidean row norm.

4.2 Convexity, derivatives, and smoothness

Theorem 4.3 (Effective entropic objective)**.** *Suppose each ℓ_i is convex on the prediction domain, has gradient norm at most G_0, and has Hessian bounded above by $L_0 I$. Then F_τ is convex. In the empirical inner product its gradient is*

$$q_i = n p_i^* \nabla \ell_i(z_i).$$

If $n \max_i p_i^* \le \kappa$ throughout the domain, it is smooth with

$$L_\tau \le \kappa \left(L_0 + \frac{G_0^2}{\tau} \right).$$

One may take $\kappa = \min\{n, \exp(B/\tau)\}$ when losses lie in $[0, B]$.

Proof. Equation (2) expresses F_τ as a supremum of convex functions of z, so it is convex. Differentiation gives the displayed gradient. In a direction v, direct differentiation again gives

$$\begin{aligned} D^2 F_\tau(z)[v,v] &= \sum_i p_i^* D^2 \ell_i(z_i)[v_i, v_i] \\ &\quad + \tau^{-1} \operatorname{Var}_{p^*}(\nabla \ell_i(z_i) \cdot v_i) \\ &\le \left(L_0 + G_0^2/\tau\right) \sum_i p_i^* \|v_i\|_2^2 \\ &\le \kappa \left(L_0 + G_0^2/\tau\right) \|v\|_n^2. \end{aligned}$$

The Hessian is nonnegative, and integrating its upper bound along segments proves smoothness. The same calculation handles vector outputs. □

The factor κ must not be dropped. The gradient couples different examples through their Gibbs weights. A bound independent of concentration would generally be false. The finite-sample choice $\kappa = n$ is always available even when the exponential bound is large.

For square loss, $B = (R+Y)^2/2$ suffices. For binary logistic loss, $B = \log(1+e^R)$. For C-class cross entropy, $B = 2R + \log C$ suffices. The constants in Table 1 follow from the scalar derivatives and, in the multiclass case, the covariance matrix $\operatorname{diag}(p) - pp^\top$, whose spectral norm is at most 1/2.

As $\tau \to \infty$, expansion of the finite log moment-generating function gives $F_\tau = \bar{e} + \operatorname{Var}(e)/(2\tau) + O(\tau^{-2})$. As $\tau \downarrow 0$, the maximum-error bounds imply $F_\tau \to \max e_i$. For finite τ, this paper optimizes the stated intermediate objective, not the arithmetic mean under another name.

4.3 Optional entropy over branch choices

Entropy can also be imposed on the branch oracle. Let the signed vertices be $v_1, \dots, v_N$ with $N = 2MC$, and put $s_j = \langle q, v_j \rangle_n$. For $\sigma > 0$ choose

$$r_j = \frac{e^{-s_j/\sigma}}{\sum_l e^{-s_l/\sigma}}.$$

This is the unique maximum-entropy regularized minimizer of $\sum_j r_j s_j - \sigma \mathsf{H}(r)$. If one vertex is sampled from r, its expected oracle defect is at most $\sigma \log N$.

Proof. At the regularized minimum, comparison with a point mass at a minimizing vertex gives $\sum_j r_j s_j - \sigma \mathsf{H}(r) \le \min_j s_j$. Since $\mathsf{H}(r) \le \log N$, rearrangement proves the claim. □

Sampling adds one branch per update, whereas taking the Gibbs mean of all vertices can add the whole dictionary at once. The expected convergence theorem below applies with oracle defect $\sigma \log N$. Our principal comparisons use maximum-entropy selection of examples and an exact maximizing-correlation branch oracle; this isolates the user's requested error-selection strategy from additional branch-sampling variance.

5 Effective fitting and its certificates

5.1 The exact linear oracle

For any empirical gradient q, an exact linear minimizer on $\mathcal{K}_{R,k}$ is obtained by

$$j^* \in \arg\max_j |\langle q, A_j \rangle_n|, \qquad s = -R \operatorname{sign}(\langle q, A_{j^*} \rangle_n) A_{j^*}.$$

At zero correlation either sign is acceptable. In the multiclass case maximize over the atom and class coordinate. The oracle is a finite scan, so its mathematical existence is accompanied by an explicit algorithm.

Definition 5.1 (Oracle and gradient errors)**.** An approximate oracle returns $s \in \mathcal{K}$ satisfying

$$\langle \widetilde{q}, s \rangle_n \le \min_{v \in \mathcal{K}} \langle \widetilde{q}, v \rangle_n + \delta.$$

A gradient approximation satisfies $\|\widetilde{q} - \nabla F(z)\|_n \le \eta$. Put $e = \delta + 2R\eta$.

Lemma 5.2 (Transfer of oracle accuracy)**.** *An oracle as above is an e-accurate linear oracle for the true gradient. Moreover*

$$F(z) - F^* \le \widehat{G}(z) := \langle \widetilde{q}, z - s \rangle_n + \delta + 2R\eta.$$

Proof. Let v minimize the true linear objective. Both s, v are within distance R of zero. Replacing $\widetilde{q}$ by $\nabla F(z)$ changes the comparison of s and v by at most $\eta \|s - v\|_n \le 2R\eta$. Convexity gives $F(z) - F(z^*) \le \langle \nabla F(z), z - z^* \rangle_n$. Compare the approximate oracle with z^* and use the same diameter bound. □

5.2 Recursive entropic boosting

The following procedure uses $F = F_\tau$. The identical oracle and proof apply to an ordinary smooth average loss.

Algorithm 1: polynomial recursive entropic boosting.

1. Generate canonical gate multisets of size at most k, cache their sample products, and form the cost-normalized atom matrix A.
2. Set $\theta_0 = 0$ and $z_0 = 0$.
3. At round t, compute the losses, maximum-entropy weights p_i^* from (3), and gradient $q_{t,i} = np_i^*\nabla\ell_i(z_{t,i})$.
4. Find the signed atom s_t with the best linear score, exactly or with certified defect δ_t.
5. Compute the gap from Lemma 5.2. Stop if it is at most the requested tolerance.
6. Otherwise update $z_{t+1} = (1-\gamma_t)z_t + \gamma_t s_t$ and the corresponding coefficients, using $\gamma_t = 2/(t+2)$ or any step whose objective is no larger than that scheduled step.

Exact line search is allowed by the last clause. The numerical comparison code uses an Armijo candidate safeguarded by evaluating the scheduled candidate and taking the better one. This ensures that the proof's descent bound applies even though the actual step need not equal the schedule. Extra loss evaluations used by this safeguard are recorded separately.

Theorem 5.3 (Fitting rate with explicit defects)**.** *Let F be convex and L-smooth on $\mathcal{K}_{R,k}$. Suppose all gradient/oracle defects obey $\delta_t + 2R\eta_t \le e$. For Algorithm 1 and every $T \ge 1$,*

$$F(z_T) - F^* \le \frac{8LR^2}{T+2} + e. \tag{5}$$

Every iterate is feasible and has at most T active signed atom coordinates. For the Gibbs example selector one may use $L = L_\tau$ from Theorem 4.3. For a sampled entropic branch oracle, the same bound holds in expectation with $e \ge \sigma\log(2MC) + 2R\eta$.

Proof. The diameter of $\mathcal{K}$ in the empirical norm is at most $2R$. Thus its curvature constant is bounded by $C_F \le 4LR^2$, meaning

$$F(z+\gamma(s-z)) \le F(z) + \gamma\langle\nabla F(z), s-z\rangle_n + \tfrac{1}{2}C_F\gamma^2.$$

Let $h_t = F(z_t) - F^*$. Convexity and Lemma 5.2 give

$$h_{t+1} \le (1-\gamma_t)h_t + \gamma_t e + \tfrac{1}{2}C_F\gamma_t^2.$$

This is valid for the scheduled step, hence also for a step with a smaller objective. Since $\gamma_0 = 1$, $h_1 \le e + C_F/2$. Induction gives $h_t \le e + 2C_F/(t+2)$: substituting the bound on h_t gives $e + 2C_F(t+1)/(t+2)^2 \le e + 2C_F/(t+3)$. Insert $C_F \le 4LR^2$.

Feasibility follows from convexity of $\mathcal{K}$. Each update scales old coefficients and adds one signed coordinate.

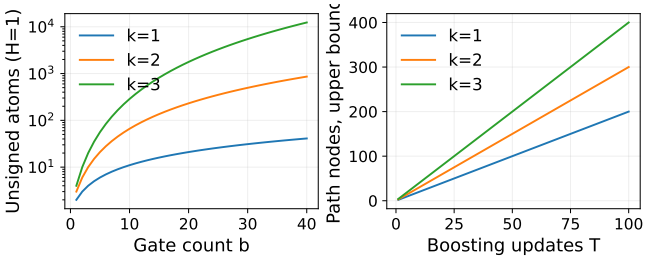


Figure 2: Exact candidate counts $\binom{b+k}{k}$ for one leaf, and the path-node bound $(k+1)T$. These are mathematical counts, not measured runtimes.

Combining equal labels can only reduce support. For a sampled oracle, condition on the current iterate, replace the oracle defect by its conditional expectation, and repeat the same induction for expected suboptimality. □

Corollary 5.4 (Precision and branch budget)**.** *For $\varepsilon > e$, any integer $T \ge 1$ satisfying $T+2 \ge 8LR^2/(\varepsilon - e)$ ensures empirical suboptimality at most ε. A model with this guarantee has at most T active coordinates, with at most kT gate nodes before prefix sharing and at most $T-1$ binary addition nodes. Including one coefficient node and one leaf per term gives at most $(k+3)T - 1$ nodes.*

This is a polynomial bound in the numerical accuracy parameter $1/(\varepsilon - e)$, not a logarithmic-precision guarantee. It also depends on the radius, temperature, and concentration through L. An ill-conditioned task can require many updates even when the candidate dictionary is small.

Proposition 5.5 (General polynomial insertion schedule)**.** *If round t inserts at most $C(t+1)^p$ branches, with $p \ge 0$, and every inserted branch has degree at most k, then after T rounds its expanded branch count is at most CT^{p+1} and its unshared path-node count is at most $(k+1)CT^{p+1}$. This statement requires insertion and coefficient rescaling, rather than repeated substitution into every existing leaf.*

Proof. Sum the insertion bounds and use $(t+1)^p \le T^p$ for $0 \le t < T$. Each branch has at most k gates and one leaf. Coefficient rescaling does not create a new path. □

5.3 Preprocessing, runtime, and bit complexity

Generating the $\binom{b+k}{k}$ canonical gate products costs $O(n\binom{b+k}{k})$ arithmetic operations using one cached parent product for each new word. Multiplying by the H leaves forms A in $O(nM)$ operations. A complete exact scalar oracle costs $O(nM)$ per round; a C-class oracle costs $O(nMC)$. The full matrix occupies $O(nM)$ scalars. Prefix products and streamed columns can trade repeated arithmetic for lower storage.

With Q_T extra full-sample loss evaluations for line search, the scalar arithmetic work is $O(nM + TnM + Q_T n)$;

the corresponding vector expression includes C in loss and oracle work. This accounting is why equal numbers of full gradient passes do not by themselves prove equal computational work.

Theorem 5.6 (Effective specialization)**.** *For rational gate values, rational bounded leaves, rational labels, and rational R, square-loss fitting with the fixed schedule is an exact rational algorithm. At fixed k, its arithmetic operation count is polynomial in n, H, b, T, and its bit complexity is polynomial in these parameters and the input encoding length. For computable loss and gate data, approximate fitting with prescribed positive gradient and oracle tolerances is effective.*

Proof. Every atom contains at most k products of rational gate values and one rational normalization. Its encoding length is polynomial in the input length and k. All square-loss gradients, inner products, comparisons, and fixed steps are rational. After T updates starting at zero, the weight of the signed vertex chosen at update j, with $0 \le j < T$, is

$$w_{j,T} = \frac{2(j+1)}{T(T+1)}.$$

The identity follows by telescoping the later factors $(1 - 2/(t+2))$. Accumulate signed integers $j+1$ per coordinate and apply the common denominator only when forming predictions. These integers have $O(\log T)$ bits beyond the magnitude bound $O(T^2)$. Rational dot products involve only polynomially many bounded-length denominators, whose product still has polynomial encoding length. Standard integer arithmetic therefore gives a polynomial bit bound.

For computable inputs, each of finitely many scores can be approximated within a requested tolerance. Selecting the smallest approximate score incurs at most twice its score error, so no exact equality test on arbitrary computable reals is needed. Stable log-sum-exp computes the entropic weights; bounded derivative approximations supply η. The proof invokes no oracle for noncomputable labels. □

The exact implementation `exact_square.py` uses only Python's rational arithmetic. The large-sample numerical experiments use floating-point arithmetic. Their reported gaps are numerical diagnostics, rather than outward-rounded interval proofs. Lemma 5.2 specifies how a certified numerical implementation would inflate those gaps.

5.4 Pruning and empirical compression

An exact oracle need not score every continuation when a valid upper bound rules it out. If v_i is the product of the gates in a prefix, every continuation has additional factors in $[0, 1]$ and a leaf bounded by one. If its normalization cost is at least $c_{\min}$, then

$$R\left|\frac{1}{n}\sum_i q_i a_i\right| \le \frac{R}{n c_{\min}} \sum_i |q_i| v_i. \tag{6}$$

Comparing this bound to the best completed score permits branch-and-bound pruning. It preserves exactness but need not improve the worst case. It also bounds the contribution of all unvisited descendants, so an early stop can yield a certified approximate oracle rather than an unqualified claim that no better branch exists.

Proposition 5.7 (Sample-preserving support reduction)**.** *Any feasible empirical prediction in $\mathbb{R}^{nC}$ has a representation by at most $nC+1$ signed vertices. A supplied rational representation can be reduced to this size by rational linear algebra in polynomial time in its supplied support size and dimension.*

Proof. If more than $nC+1$ vertices have positive convex weights, the augmented vectors $(v, 1)$ are linearly dependent. Choose a nonzero dependence λ with $\sum \lambda = 0$ and $\sum \lambda v = 0$. It has positive and negative components. Subtract $t\lambda$ from the weights with $t = \min_{\lambda_j > 0} w_j/\lambda_j$. The new weights are nonnegative, preserve the prediction and total weight, and set at least one positive weight to zero. Repeat. Gaussian elimination and rational weight updates are effective; determinants of bounded rational matrices give polynomial encoding bounds. □

This compression preserves the training predictions. It can change predictions away from the sample because distinct functions may have equal sample columns. We do not apply this sample-only compression before reporting held-out performance.

6 SGD for precisely the same entropic objective

6.1 A separable auxiliary formulation

Computing a Gibbs distribution only inside each random minibatch would optimize a batch-dependent log-sum-exp surrogate. To avoid changing the comparison problem, introduce one scalar a and define

$$\Phi(\theta, a) = a + \frac{\tau}{n}\sum_i \exp\left(\frac{\ell_i((A\theta)_i) - a}{\tau}\right) - \tau. \tag{7}$$

Theorem 6.1 (Exact stochastic reduction)**.** *The function Φ is jointly convex in (θ, a). For every θ,*

$$\min_{a \in \mathbb{R}} \Phi(\theta, a) = F_\tau(A\theta), \qquad a^*(\theta) = F_\tau(A\theta).$$

If all losses lie in $[0, B]$, restriction to $a \in [0, B]$ leaves the minimum unchanged. For a uniformly sampled example I, put

$$w_I = \exp((\ell_I - a)/\tau).$$

Then $w_I \nabla_\theta \ell_I$ and $1 - w_I$ are unbiased stochastic gradients of the two coordinates of Φ.

Proof. Each exponential is a convex nondecreasing function applied to the convex expression $(\ell_i(A\theta) - a)/\tau$. Adding a linear function preserves joint convexity. Differentiating with respect to a gives

$$\partial_a \Phi = 1 - \frac{1}{n} \sum_i e^{(\ell_i - a)/\tau}.$$

It vanishes uniquely at $a = \tau \log(n^{-1} \sum_i e^{\ell_i/\tau})$. At that value the exponential average equals one and (7) equals a. The bounds in Proposition 4.2 put the minimizer in $[0, B]$. Averaging the displayed sample gradients over a uniform index gives the two derivatives of (7) exactly. □

Algorithm 2: auxiliary-objective SGD. Sample one example or a uniformly sampled minibatch with replacement. Average the stochastic gradients from Theorem 6.1; take a gradient step in θ and a; project θ in Euclidean distance onto $\|\theta\|_1 \le R$ and clip a to $[0, B]$. Average the iterates for output. This is stochastic gradient descent for a convex objective with exactly the same optimized value as Algorithm 1.

Proposition 6.2 (Finite stochastic guarantee)**.** *Let $\mathcal{D} = \{\|\theta\|_1 \le R\} \times [0, B]$, with Euclidean diameter at most $D = \sqrt{4R^2 + B^2}$. Suppose stochastic gradients are unbiased and have squared norm expectation at most G^2. Projected SGD with constant step $\eta = D/(G\sqrt{S})$ and the average of S iterates satisfies*

$$\mathbb{E}\big[F_\tau(A\bar{\theta}) - F_\tau^*\big] \le \frac{DG}{\sqrt{S}}.$$

Such a finite G exists for the bounded dictionary: one can use

$$G^2 \le e^{2B/\tau} M G_0^2 + (1 + e^{B/\tau})^2.$$

Proof. Euclidean projection is nonexpansive. For $u^* \in \arg\min_{\mathcal{D}} \Phi$,

$$\|u_{s+1} - u^*\|^2 \le \|u_s - u^*\|^2 - 2\eta \langle \widehat{g}_s, u_s - u^* \rangle + \eta^2 \|\widehat{g}_s\|^2.$$

Take conditional expectations, use convexity, sum, and telescope. Jensen's inequality bounds the objective at the average by the average objective, giving $D^2/(2\eta S) + \eta G^2/2$. The selected step gives $DG/\sqrt{S}$. Since $F_\tau(A\bar{\theta}) = \min_a \Phi(\bar{\theta}, a) \le \Phi(\bar{\theta}, \bar{a})$, the same bound applies to the reduced risk. Finally each atom row has Euclidean norm at most $\sqrt{M}$, the loss gradient norm is at most G_0, and $w \le e^{B/\tau}$. Bounding both gradient coordinates gives the stated G. □

The worst-case stochastic constant can be large. It is not evidence that practical SGD must be slower. Conversely the $O(1/T)$ full-gradient rate does not imply greater sample efficiency than the $O(1/\sqrt{S})$ stochastic bound: one boosting round inspects the whole sample, whereas one SGD step need not. The experiments compare matched example-gradient counts and disclose additional objective work and rate tuning.

7 Statistical interpretation

An empirical optimization certificate controls fitting error inside the chosen class. It does not itself control population error. This distinction matters especially when the objective deliberately tilts toward the largest training losses.

Consider scalar predictors $f = \sum_j \theta_j a_j$ with $\|\theta\|_1 \le R$ and a dictionary fixed independently of the training sample. Suppose the examples are independent and identically distributed, $0 \le \ell(y, f(x)) \le B$, and the loss is K_0-Lipschitz in its prediction on $[-R, R]$. Define

$$\mathcal{R}_\tau(f) = \tau \log \mathbb{E} \exp(\ell(Y, f(X))/\tau).$$

This is the population counterpart of the empirical entropic objective, and is distinct from the population arithmetic mean loss.

Lemma 7.1 (Complexity of the recursive hull)**.** *For any fixed sample, the empirical Rademacher complexity of the scalar prediction class is at most*

$$R\sqrt{\frac{2\log(2M)}{n}}.$$

Proof. The supremum of a linear functional over the coefficient ℓ^1 ball is R times the maximum absolute atom correlation. For each signed atom, the Rademacher sum has a sub-Gaussian moment-generating bound with variance proxy $1/n$, since $|a_j(x_i)| \le 1$. A union bound in exponential-moment form over $2M$ signed atoms gives expected maximum at most $\sqrt{2\log(2M)/n}$. Multiply by R. □

Theorem 7.2 (One bounded population guarantee)**.** *Under the preceding scalar hypotheses, with probability at least $1 - \delta$, simultaneously for all predictors in the class,*

$$|\mathcal{R}_\tau(f) - F_{\tau,D}(f)| \le \Gamma_n,$$

$$\Gamma_n = 4K_0 e^{B/\tau} R\sqrt{\frac{2\log(2M)}{n}} + 3\tau e^{B/\tau}\sqrt{\frac{\log(4/\delta)}{2n}}.$$

Consequently, a fitted predictor with empirical suboptimality at most ε satisfies

$$\mathcal{R}_\tau(\widehat{f}) \le \inf_{f \in \mathcal{F}} \mathcal{R}_\tau(f) + \varepsilon + 2\Gamma_n.$$

Proof. The transformed loss $g_f = \exp(\ell_f/\tau)$ is bounded between 1 and $e^{B/\tau}$ and has prediction Lipschitz constant at most $K_0 e^{B/\tau}/\tau$. The usual symmetrization, scalar

contraction, and bounded-differences argument [10], applied to both signs and Lemma 7.1, gives a simultaneous deviation bound

$$|\mathbb{E}g_f - P_n g_f| \le \frac{4K_0 e^{B/\tau} R}{\tau}\sqrt{\frac{2\log(2M)}{n}} + 3e^{B/\tau}\sqrt{\frac{\log(4/\delta)}{2n}}.$$

The constants are conservative. Both arguments of the logarithm are at least one; the map $v \mapsto \tau \log v$ is therefore τ-Lipschitz on their range. Multiplication by τ yields Γ_n. Apply the uniform bound to the fitted predictor and to a population comparator, with the empirical optimization inequality between them, and then take the infimum. □

For fixed k, $\log M$ grows only as $\log H + k\log(b+k)$ up to constants. The exponential temperature factor can still make this generic bound uninformative. The theorem is an existence and separation result, not a tight explanation of the small-data experiments. Better localized or problem-specific estimates would require additional work.

If gates, leaves, or preprocessing are selected using the same training labels, the fixed-class proof does not apply without further control. Independent dictionary construction or sample splitting restores its literal hypotheses. The experiments fit preprocessing on the training partition for ordinary predictive evaluation and do not claim that this particular statistical theorem directly certifies those data-dependent feature maps. The scalar theorem is not silently reused as a dimension-free multiclass bound.

8 What polynomial growth does and does not buy

Proposition 8.1 (Joint degree and gate growth)**.** *The complete canonical dictionary is not uniformly polynomial when both b and k are input parameters. In particular, with $b = k$, its size is at least $H2^k$.*

Proof. Every subset of a k-element gate set gives a distinct exponent vector of total degree at most k. There are 2^k such labels. The complete multiset dictionary contains all of them. □

This does not contradict the fixed-degree theorem. It also differs from the growth of ordered recursion: ordering gate words can produce b^k paths before scalar commutativity is used. For fixed b, the canonical multiset count is polynomial in k; the unrestricted joint-parameter count is the relevant obstruction here.

A DAG can share prefix products and reduce execution work while the fully unrolled tree remains large. We distinguish the stored DAG nodes, the number of expanded branch terms, and the number of candidate labels. None of these numbers is substituted for another in the complexity claims.

Finally, convex-hull optimization is not a solution of arbitrary fixed-support selection. A gap certificate compares with every coefficient vector in the full feasible hull, which is a useful and stronger numerical comparator for the objective, but the method need not return the sparsest optimum or the optimum attainable under a smaller prescribed support cap. A chosen early stopping budget gives the approximation bound in Corollary 5.4.

9 Undergraduate examples and controlled comparisons

9.1 Common experimental contract

The comparison suite contains six examples commonly used when teaching supervised learning: synthetic linear regression, XOR, two moons, binary logistic regression on the breast-cancer dataset, three-class Iris classification, and ten-class handwritten digit classification. The latter three use the small bundled scikit-learn datasets documented in [12]. Digits is the 1797-example 8×8 dataset, not MNIST.

For each dataset and seed $11, 29, 47$, observations are split into approximately 60 percent training, 20 percent validation, and 20 percent test data. Classification and XOR splits are stratified. Standardization, any PCA map, and min–max gate scaling are fitted on training observations only. Inputs with more than eight features are reduced to eight principal components; this affects breast cancer and digits, and the retained variance is recorded per run. Values outside the training gate range are clipped to $[0, 1]$ on validation and test data. Regression targets, except signed XOR, are standardized using training target moments.

The leaves consist of the constant one, and $c_\alpha = 1 + |\alpha|$. Degree-one and degree-two dictionaries include the same lower-order atoms. The norm radii are 24 for linear regression and XOR, 40 for two moons and breast cancer, 60 for Iris, and 120 for Digits. All entropic comparisons use $\tau = 1$. Each method uses 200 passes worth of training-example gradient evaluations per fitted candidate. A boosting pass computes a full gradient and a complete atom scan; SGD-1 draws n independent uniform examples per pass; SGD-32 draws minibatches with replacement whose total size is n per pass.

The SGD baseline is Algorithm 2, with both the coefficient and auxiliary scalar updates. It uses $\eta_e = \eta_0/\sqrt{1+e}$ in epoch e and averages coefficient iterates over the final half of training. SGD-1 selects $\eta_0 \in \{0.003, 0.03, 0.3\}$ by validation entropic risk. SGD-32 uses the wider grid $\{0.003, 0.03, 0.3, 3, 30\}$ because its per-pass number of updates is smaller. The validation search never uses test loss. The individual records include invalid numerical candidates and total tuning time. The finite-horizon guarantee in Proposition 6.2 establishes the common objective and an available guaranteed schedule; it is not a claim

Task	Boost	SGD-1	SGD-32
Linear regression	0.016	0.019	0.016
XOR	0.013	0.010	0.146
Two moons	0.416	0.414	0.414
Breast cancer	0.324	0.306	0.283
Iris	0.353	0.292	0.327
Digits	1.926	1.555	1.568

Table 2: Mean held-out entropic risk over three splits, degree two, temperature one. Lower is better. All methods use the same atoms, radius, and 200 training-example gradient passes. SGD rates are selected on validation data.

Task	Boost	SGD-1	SGD-32
Two moons	87.7	87.3	87.5
Breast cancer	90.9	92.7	93.3
Iris	96.7	96.7	97.8
Digits	59.8	78.6	79.3

Table 3: Mean held-out accuracy (percent), with the same degree-two comparison settings as Table 2. Iris uses three-class softmax and Digits uses ten-class softmax.

that these empirically tuned decreasing rates equal that theoretical schedule.

Boosting uses a complete maximizing-correlation oracle. Its step is the better of a backtracking candidate and the scheduled conditional-gradient candidate. The extra full-sample loss evaluations are stored under `extra_sample_losses`. Thus gradient counts are matched, while objective-evaluation work is not claimed equal. Features are cached for every optimizer. All runs use one numerical thread. Reference Python timings include training diagnostics; the initial comparison run and extended minibatch search overlapped in execution. Timing differences are therefore exploratory and do not establish optimized implementation speedups.

9.2 Linear regression

The synthetic task samples $x \in [-1,1]^6$ and

$$y = x \cdot (0.4, 0.8, -0.7, 0.2, 0.1, -0.3) + \epsilon, \quad \epsilon \sim N(0, 0.1^2).$$

There are 800 observations before splitting. With degree one, the model is a constrained affine linear regressor after the gate map. Degree two adds interactions without changing the data or the loss. For square loss the error-gradient contribution is $np_i^*(z_i - y_i)$, so the entropy selector emphasizes squared residuals while the signed residual still determines the direction of correction. A better held-out degree-one result can be expected when added interactions only increase finite-sample variance or optimization burden.

Task / method	Branches	Seconds	Gap
Linear regression / Boost	7.0	0.150	0.099
Linear regression / SGD-1	28.0	3.023	0.109
Linear regression / SGD-32	28.0	0.190	0.035
XOR / Boost	4.0	0.087	0.190
XOR / SGD-1	6.0	2.738	0.120
XOR / SGD-32	6.0	0.101	0.676
Two moons / Boost	3.3	0.086	0.136
Two moons / SGD-1	6.0	2.914	0.106
Two moons / SGD-32	6.0	0.125	0.012
Breast cancer / Boost	4.3	0.091	0.280
Breast cancer / SGD-1	45.0	3.085	0.072
Breast cancer / SGD-32	45.0	0.197	0.044
Iris / Boost	4.3	0.192	0.379
Iris / SGD-1	15.0	4.399	0.157
Iris / SGD-32	15.0	0.173	0.251
Digits / Boost	6.3	0.586	2.350
Digits / SGD-1	45.0	62.080	0.113
Digits / SGD-32	45.0	2.108	0.134

Table 4: Degree-two active branches, selected-run Python fitting time, and numerical training gap, each averaged over three splits. Time includes training diagnostics, excludes feature construction and unselected tuning trials, and is exploratory rather than an optimized solver comparison.

Task	B1	S1	B2	S2
Linear regression	0.016	0.015	0.016	0.016
XOR	0.500	0.500	0.013	0.146
Two moons	0.416	0.414	0.416	0.414
Breast cancer	0.324	0.276	0.324	0.283
Iris	0.367	0.384	0.353	0.327
Digits	1.928	1.562	1.926	1.568

Table 5: Representation comparison: B1/B2 are boosting at degree one/two, S1/S2 are minibatch SGD at degree one/two. The cost of higher degree is charged by the same radius constraint. A higher degree need not improve a finite-budget held-out result.

9.3 XOR and recursive feature interactions

The 800-example XOR dataset repeats the four Boolean input pairs equally. Example 2.6 gives an exact recursive representation and a proof of the degree-one approximation barrier. Here all optimizers at degree two receive the same interaction feature $g_1 g_2$ and the same radius. Consequently the optimizer comparison does not credit boosting with a feature withheld from SGD. Degree-one entropic risk is also at least $1/2$ on the balanced distribution: it is no smaller than mean square loss, whose optimum is $1/2$, and zero attains $1/2$ with equal losses.

The separate rational reference run uses the four distinct Boolean inputs, the fixed conditional-gradient schedule, $R = 24$, and 20000 updates. Its exact rational objective and gap are included in the machine-readable evidence. This deliberately checks the theorem-prescribed arithmetic rather than substituting the much faster numerical line-search variant for a rational proof.

9.4 Binary logistic regression and two moons

For labels $y_i \in \{-1, 1\}$,

$$\ell_i(z_i) = \log(1 + e^{-y_i z_i}), \qquad \ell_i'(z_i) = -y_i \operatorname{sigmoid}(-y_i z_i).$$

The Gibbs selector acts on the logistic loss, not on the unsigned raw classification error. This gives a smooth selection strategy even when the predicted class changes discontinuously. The breast-cancer task is an ordinary binary logistic example with an eight-dimensional training-fitted representation. It is a benchmark calculation, with no claim about clinical deployment.

Two moons uses 800 generated points with noise standard deviation 0.18. Its curved classes illustrate the limits of a low-degree feature map. Degree two does not represent every curved decision boundary, and a larger dictionary need not improve the result within 200 updates and the stated radius.

9.5 Iris and handwritten digits

For C classes, scores are $z_i \in \mathbb{R}^C$ and

$$\ell_i(z_i) = \log \sum_{c=1}^{C} e^{z_{ic}} - z_{i,y_i}.$$

Its gradient is $\operatorname{softmax}(z_i) - e_{y_i}$. The entropic weights multiply this full vector. The signed branch oracle maximizes the magnitude of its atom correlation over all class coordinates, and therefore adds one atom–class coefficient per round. SGD uses the full coefficient matrix and the identical total ℓ^1 radius.

Iris has 150 observations and four original features; Digits has 1797 observations and 64 pixels before the eight-component PCA map. The small Iris split has appreciable accuracy granularity. Digits is a stronger test of allocating a finite branch budget among multiple classes. A low active branch count accompanied by lower accuracy is a compression–performance tradeoff, not evidence of an optimization advantage.

9.6 Mean-loss controls and implementation checks

We additionally run four control families for ordinary average loss: balanced XOR, noisy binary-gate regression, noisy soft-gate regression including g_2^2, and a binary logistic interaction problem. These are separate objectives from the maximum-entropy comparisons. The controls test recursive evaluation, repeated soft gates, and the representation effect of adding degree-two branches.

The verification scripts check direct recursive versus canonical evaluation, exact dictionary counts, a valid prefix-pruning bound, the rational XOR identity, and a numerical primal-gap inequality against an independent full-hull SLSQP solution. They also check finite-difference loss gradients, the Gibbs variational identity, the exact auxiliary-objective identity, the unbiased estimator at the optimal auxiliary scalar, monotonic safeguarded descent, feasibility, and the KKT conditions of the coefficient projection. These are executable implementation checks supplementing the mathematical proofs; they are not a substitute for those proofs.

Task	k	Atoms	Test loss
XOR	1	3	0.50000
XOR	2	6	0.00001
Binary gates	1	7	0.18705
Binary gates	2	28	0.00131
Soft gates	1	6	0.01646
Soft gates	2	21	0.00049
Logistic	1	5	0.45833
Logistic	2	15	0.40672

Table 6: Separate average-loss control experiments, not entropic-objective comparisons. XOR uses the balanced four-point distribution; the other rows average three random seeds. The budget is at most 4000 updates.

9.7 Observed outcomes and their limits

The lowest mean held-out entropic risk is attained by an SGD variant on each of the six tasks. Boosting does not establish predictive superiority in this comparison. It consistently uses fewer active branch functions: for example, the degree-two models use about 7 versus 28 branches on linear regression, 4 versus 6 on XOR, and about 6 versus 45 on Digits. The coefficient-coordinate count for multiclass predictions is recorded separately from the number of distinct branch functions.

The difference is material on Digits: mean accuracy is about 59.8 percent for boosting, 78.6 percent for SGD-1 and 79.3 percent for SGD-32. The sparse 200-update boosting model is underfit at this budget; its mean numerical training gap is about 2.35. Thus the experiment does not compare three nearly converged solutions. On Iris, the corresponding accuracies are about 96.7, 96.7 and 97.8 percent. Three splits are sufficient for a reproducible illustration, not for broad performance ranking or a significance claim.

Broadening the minibatch learning-rate grid substantially improves the SGD-32 results on several tasks. Both the initial and extended tuning records are supplied, so the published comparison uses the stronger validation-selected baseline rather than the initially slower small-rate setting. Every invalid candidate is recorded with its numerical error. No test result is used to choose the rate.

The maximum-entropy distribution remains quite diffuse at the chosen temperature. Across the degree-two boosting runs, the smallest entropy effective-count fraction is 0.910. This reports actual selection entropy rather than assuming concentrated attention to a small error subset. Temperature sweeps and adversarially corrupted-label studies are outside this experiment.

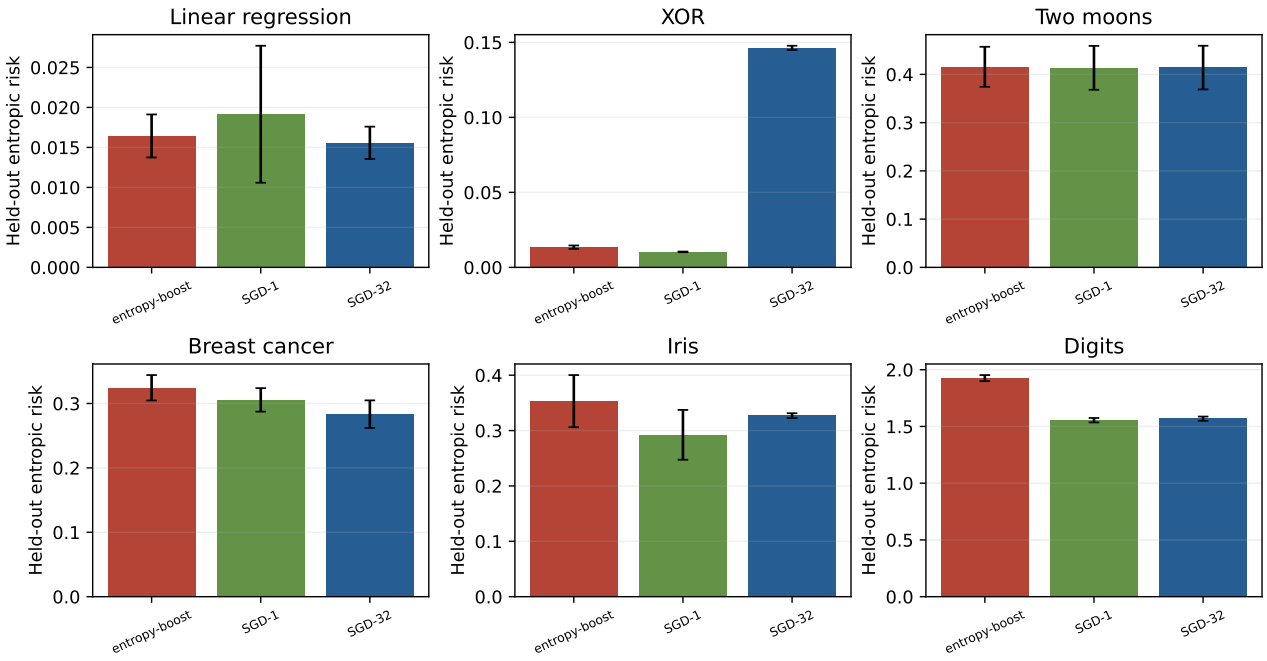


Figure 3: Degree-two entropic-risk comparisons on the six undergraduate tasks. Error bars are sample standard deviations over three train/validation/test splits. Each SGD rate is selected on the validation split. Numerical values and full per-run records are included in the tables and companion archive.

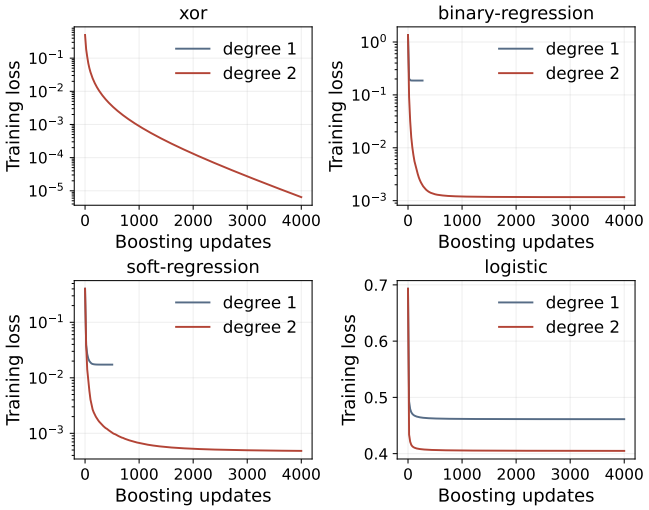


Figure 4: Average-loss controls at degrees one and two. The non-XOR curves use seed 11. These illustrate fitting behavior; they do not compare the two different objectives.

Matching 200 example-gradient passes does not match all arithmetic. Boosting uses an average of 647.3 additional full-sample-equivalent loss evaluations in its step search and safeguard. The archive records this work and the tuning costs. Its short Python wall times reflect a vectorized full scan compared with a reference stochastic loop, and do not establish a hardware-independent speedup.

The exact rational XOR reference has square loss 0.00018774 and primal gap 0.00443325 after 20000 scheduled updates. The gap is a true rational certificate for its finite comparison hull. The ordinary-loss line-search control reaches a much smaller loss in fewer updates, illustrating the practical effect of step choice without altering the rational reference claim.

10 Interpretation, limits, and reproducibility

The construction has three mathematical layers. The free polynomial monad specifies typed recursive presentations and substitution. The canonical monomial module gives a finite dictionary with an explicit degree and resource accounting. The empirical convex hull provides an optimization domain on which entropy-selected errors and branch oracles have computable meanings. The evaluation maps connect these layers; none of the layers is identified with all of the others.

Higher categorical structure is useful for retaining symmetries, implementation equivalences, and descent data. Its image in a scalar empirical objective is 0-truncated, as proved in Proposition 3.2. In particular, nontrivial homotopy in a presentation does not supply an unexplained numerical advantage to the optimizer. Any practical advantage must come from a chosen representation, oracle implementation, approximation, or resource policy.

Proposition 10.1 (Equivariance of the empirical algorithm)**.** *Permuting the rows of the sample and atom matrix permutes the Gibbs weights and predictions, preserves the*

objective and all linear scores, and preserves the iterate coefficients when the same atom enumeration is used for ties.

Proof. A row permutation preserves the finite sum defining log-sum-exp, carries its normalized exponential weights to the correspondingly permuted vector, and preserves every empirical inner product. The branch scores, chosen index, and coefficient update are therefore unchanged. Induct on the update count. □

This is a precise naturality statement about sample relabelling. It does not assert that a preprocessing procedure with numerical tie-breaking, or an arbitrary relabelling of tied atoms, has a canonical optimizer selection.

Limits of the claim. The complete branch search is polynomial in gate count only at fixed degree. Radius and temperature affect the iteration bound. The entropic risk deliberately favors high losses and can overweight outliers. The training-gap certificate concerns the full finite convex hull, not a globally optimal neural architecture or a minimum-support program. Rational arithmetic certifies the square-loss reference run; floating-point diagnostics are identified as such. A formal proof in the mathematical exposition is not a claim of a Lean kernel verification. No such verification is asserted here.

The empirical results compare explicit implementations under a finite tuning grid. They do not prove that recursive boosting is faster, more sample efficient, or more accurate than all SGD implementations. Large learning rates that fail numerically are retained in the experiment records, rather than being silently removed. The experiments do not include external large-scale neural-network training.

Reproduction. The companion archive contains this LaTeX source, the final PDF, Python source, environment versions, exact rational output, individual comparison records, learning curves, figures, and checksums. From its root, the following commands reproduce the principal calculations:

```
python src/exact_square.py examples/xor.json
python tests/test_contracts.py
python tests/test_entropy.py
python src/experiments.py
python src/undergraduate_comparisons.py
python src/make_tables.py
latexmk -pdf manuscript.tex
```

The data loaders use the small datasets distributed with scikit-learn; synthetic examples are generated from recorded seeds. No private dataset or credential is included. Thread counts are fixed at one for the comparison scripts. Validation chooses SGD rates; the test split is used only for final evaluation. Every quoted empirical mean is generated from saved per-run records.

Authorship and automation. This anonymous manuscript was commissioned and submitted by the responsible human user. An OpenAI assistant generated the exposition, derived and reviewed the arguments, wrote the implementations, and executed the reported checks and experiments. The byline is anonymous at the submitter's direction. No independent referee review, journal acceptance, or machine-checked theorem certification is claimed by the manuscript itself.

Result. At fixed degree, recursive gated fitting has a finite polynomial dictionary, a sound higher presentation semantics, a maximum-entropy error selector, and an effective algorithm with a branch budget and a fitting certificate. The same entropic objective admits ordinary stochastic gradients through a separable auxiliary representation. This makes the representation question and the optimizer question independently inspectable.

References

[1] R. E. Schapire. The strength of weak learnability. *Machine Learning* 5, 197–227, 1990. doi:10.1007/BF00116037.

[2] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.* 55(1), 119–139, 1997. doi:10.1006/jcss.1997.1504.

[3] J. H. Friedman. Greedy function approximation: A gradient boosting machine. *Ann. Statist.* 29(5), 1189–1232, 2001. doi:10.1214/aos/1013203451.

[4] M. Jaggi. Revisiting Frank–Wolfe: Projection-free sparse convex optimization. *PMLR* 28(1), 427–435, 2013. https://proceedings.mlr.press/v28/jaggi13.html.

[5] C. Cortes, X. Gonzalvo, V. Kuznetsov, M. Mohri, and S. Yang. AdaNet: Adaptive structural learning of artificial neural networks. *PMLR* 70, 874–883, 2017. https://proceedings.mlr.press/v70/cortes17a.html.

[6] N. Gambino and J. Kock. Polynomial functors and polynomial monads. *Math. Proc. Cambridge Philos. Soc.* 154, 153–192, 2013. doi:10.1017/S0305004112000394. https://arxiv.org/abs/0906.4931.

[7] D. Gepner, R. Haugseng, and J. Kock. ∞-Operads as analytic monads. *Int. Math. Res. Not.* 2022(16), 12516–12624, 2022. https://arxiv.org/abs/1712.06469.

[8] J. Lurie. *Higher Topos Theory.* Annals of Mathematics Studies 170, Princeton University Press, 2009. Author's maintained text: https://www.math.ias.edu/~lurie/papers/HTT.pdf.

[9] T. Li, A. Beirami, M. Sanjabi, and V. Smith. On tilted losses in machine learning: Theory and applications. *J. Mach. Learn. Res.* 24(142), 1–79, 2023. https://jmlr.org/papers/v24/21-1095.html.

[10] P. L. Bartlett and S. Mendelson. Rademacher and Gaussian complexities: Risk bounds and structural results. *J. Mach. Learn. Res.* 3, 463–482, 2002. https://www.jmlr.org/papers/v3/bartlett02a.html.

[11] H. Robbins and S. Monro. A stochastic approximation method. *Ann. Math. Statist.* 22(3), 400–407, 1951. doi:10.1214/aoms/1177729586.

[12] Scikit-learn developers. Toy datasets; generated datasets; load_iris, load_digits, load_breast_cancer, and make_moons API documentation. Accessed November 10, 2026. https://scikit-learn.org/stable/datasets/toy_dataset.html. Package versions used are recorded in the companion archive.