How Training Objectives Allocate Predictive Information

Anonymous authors
Paper under double-blind review

Abstract

A representation with a fixed number of bits or a fixed linear rank must allocate information among the predictions it will support. We study how contrastive matching makes this choice when the prediction context arrives after encoding. A finite candidate list caps the gain from any one request. Matching can therefore favor information for harder requests even when prediction values it less. For two independent uniform bit strings with later complementary information, we solve prediction over all randomized encoders within one string's bit length and bound unavoidable predictive regret from attained matching scores. In two of six controlled neural settings, matching training loses information from a supplied prediction-optimal code. Simultaneous risk bounds at 95% confidence for the fitted models show that a smoothed direct predictor outperforms every predictor of the paired matching code. Withholding complementary information reverses the minimum-risk ordering for the same parity-initialized pairs. We then characterize every optimal linear matching subspace in a Gaussian block model and quantify how predictive cost depends on optimization accuracy. Uncapped power divergences can favor a less frequent, more reliably observed string. Bounds over all randomized fixed-bit encoders distinguish bounded from growing regret as divergence order approaches one and string length grows. These results separate objective-induced information loss from errors in fitting a predictor.

1 Introduction

Contrastive learning trains representations by distinguishing related observations from alternatives (van den Oord et al., 2018; Chen et al., 2020). CLIP, for example, learns from image–text pairs and transfers to image classification (Radford et al., 2021). That reuse depends on which features survive training: contrastive objectives can suppress useful features (Chen et al., 2021; Robinson et al., 2021; Xue et al., 2023).

When a representation cannot retain everything useful, this becomes a question of allocation. With a fixed number of bits, or a fixed rank for a linear representation, retaining information for one prediction can leave less for another. Predictive compression studies how to preserve task information under a rate constraint (Dubois et al., 2021); recent vision-language systems also compress and cache visual tokens for later retrieval and generation (Bulat et al., 2025). We study how training values this competing information and what its attained score reveals about the best prediction possible from the code.

The value of retained information also depends on what the predictor will know later. In complementary-delivery coding, one message serves different requests, each accompanied by another source block (Kuzuoka et al., 2008). In our comparison, matching and prediction use the same representation constraint and later source observation. A *matching critic* assigns probabilities to the positive candidate's index; a *predictor* assigns probabilities to the target itself, without a candidate list. We call training for the latter *direct prediction*.

Consider two independent uniform eight-bit strings, A and B. An encoder sees both but can store only eight bits. A later request asks for A three times as often as B. Without additional information, storing A minimizes average predictive log loss. Now suppose the request usually comes with the other string: a request for A includes B, and conversely. Storing the coordinatewise parity $A \oplus B$ can be better. Together with either string, it recovers the other. If the other string is supplied on 90%

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

of requests, parity is prediction-optimal and leaves 0.8 bits of average uncertainty, all from requests without the other string.

A two-candidate matching task can favor another code. Store A_1, B_2 and six parities at the remaining positions. With the other string present, this code reveals seven requested bits instead of eight. Without it, the code still reveals one bit. That raw bit helps an otherwise ambiguous match more than the eighth parity helps a match that is already easy. The mixed code therefore has lower matching loss, even though its prediction loss is twice that of parity. Figure 1 gives the exact comparison. The mixed code need not be matching-optimal.

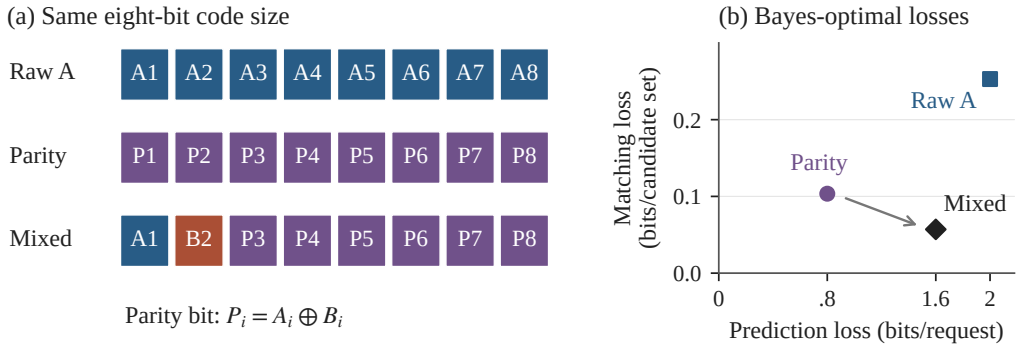

Figure 1: **The objective changes how the same eight bits are used.** The request favors A with probability .75 and supplies the other string, outside the code, with probability .9. Parity is prediction-optimal; the feasible mixed code matches better but doubles prediction loss. Losses use optimal predictors and critics, in bits per request or two-candidate set, respectively.

Contrastive transfer guarantees depend on pair generation, shared task information and the representation class (Arora et al., 2019; Tian et al., 2020). Saunshi et al. (2022) exhibit contrastive optima with poor linear transfer and derive function-class-dependent guarantees for a spectral loss; our bounds concern information unavailable to any predictor of the stored code. Recent sufficiency-based guarantees retain approximation error for the score class (Lin & Mei, 2025). Surrogate-loss theory studies equivalent representation choices and risk transfer (Nguyen et al., 2009; Duchi et al., 2018); contrastive experimental design controls losses from finite-candidate information estimates (Foster et al., 2020).

We solve prediction over all randomized fixed-bit encoders and bound every predictor's regret from attained matching scores. Controlled neural fits separate information loss from predictor error. The Gaussian analysis characterizes every optimal linear subspace and the role of optimization accuracy. All-encoder bounds for uncapped power divergences identify a separate allocation mechanism, distinguishing bounded from growing regret.

2 Encoding before the prediction context arrives

An encoder observes Y and stores Z before the context V of a later request arrives. A predictor assigns a distribution to the target X conditional on (Z, V) and incurs log loss, a proper scoring rule (Gneiting & Raftery, 2007). The minimum expected log loss over predictors, or Bayes risk, is

$$R(Z) = H(X \mid V, Z) = H(X \mid V) - I(X; Z \mid V).$$

For continuous targets the corresponding identity uses differential entropy. A fitted predictor can incur additional loss: even invertible codes can differ in linear prediction quality (Tschannen et al., 2020). Logarithms are natural and losses are in nats unless bits are explicitly specified.

The budget limits the stored code, not encoder or predictor parameters. Predictors and matching critics access that stored output. Reusing features before a contrastive projection head (Chen et al., 2020; Bolya et al., 2025) changes this condition: constraining the final output need not constrain those features.

For the finite-bit analysis, write the independent uniform n-bit strings as U_1, U_2 (A, B in the introduction), and put $Y = (U_1, U_2)$. The code Z takes at most 2^n values; the encoder may be randomized.

After encoding, a request J chooses string 1 with probability $p > 1/2$, and a mask S independently retains each requested coordinate with probability a. An indicator E records whether the predictor also receives the complete other string, with availability $r = P(E = 1)$. The request, mask and indicator are mutually independent and independent of the source and encoder randomness. Thus

$$X = U_{J,S}, \qquad V = (J, E, S,\ U_{3-J} \text{ when } E = 1).$$

Here $U_{J,S}$ is the part of the requested string retained by the mask. The other string is additional information outside the n-bit code budget. The target entropy without the code is $H(X \mid V) = an \log 2$.

Proposition 1 (The best code changes with later information)**.** *For every finite n, the minimum predictive risk over the stated encoder class is*

$$R^* = an \log 2\,[1 - \max(p, r)].$$

Storing U_1 attains this minimum when $r \le p$. Storing the coordinatewise parity $U_1 \oplus U_2$ attains it when $r \ge p$.

The code retains $D = I(U_1, U_2; Z)$ nats about the pair and $I_j = I(U_j; Z)$ about string j alone. Independence and the budget give $I_1 + I_2 \le D \le n \log 2$. Information about the requested masked string is at most aI_j without the other string and at most $a(D - I_{3-j})$ with it. Consequently

$$I(X; Z \mid V) \le a\{rD + (p - r)I_1 + (1 - p - r)I_2\}.$$

The maximum is $an \log 2\ \max(p, r)$, attained by the raw and parity codes. Appendix A proves the erasure inequalities.

2.1 Finite matching rewards a different use of the code

Following the candidate-classification view behind InfoNCE (van den Oord et al., 2018), we condition matching on the later context (Sordoni et al., 2021; Tsai et al., 2022). Given V, one uniformly placed positive target is drawn jointly with Z; the $K - 1$ negatives are independent draws from $P(X \mid V)$. A normalized critic sees $(Z, V, X_{1:K})$ and predicts the positive position I. Its attained population gain is

$$f = \log K + \mathbb{E} \log \widehat{p}(I \mid Z, V, X_{1:K}).$$

The gain is $\log K$ minus the matching log loss, so larger gain means better matching.

These negative targets remain uniform even given the other string. Section 4 instead uses candidate codes; both orientations obey the bound below but can have different matching values.

Identifying one candidate yields at most $\log K$ nats per request, the finite-candidate ceiling (Poole et al., 2019, Section 2.3). Prediction rewards each additional target bit available to the predictor. Every normalized critic satisfies

$$f \le \mathbb{E}_{v \sim P_V} \min\{\log K,\ I(X; Z \mid V = v)\}.$$

Once one kind of request is easy to distinguish, matching can favor information for harder requests even when prediction values it less. The cap and storage tradeoff yield a bound on predictive regret.

Proposition 2 (An attained score can force predictive regret)**.** *Let $\Delta(Z) = R(Z) - R^*$. For $p = 3/4$, $r \in \{1/2, 9/10\}$, every finite n, every randomized encoder obeying the budget, and every normalized matching critic,*

$$\Delta(Z) \ge c_r[f - t_r \log K]_+.$$

The constants are $(c_r, t_r) = (4, 7/8)$ at $r = 1/2$ and $(2, 9/10)$ at $r = 9/10$, independent of n, a and K. The bound applies to every predictor based on the code, without assuming an optimal critic or a globally optimized encoder.

For example, when the other string arrives on 90% of requests, those requests can contribute at most $.9 \log K$ to the matching gain. A larger gain must therefore use information about a string on its own. The fixed code budget makes that allocation costly for the prediction workload: each nat of gain above $.9 \log K$ forces at least two nats of predictive regret.

Appendix A gives the four-case bound for general probabilities and derives these constants. At $r = 1$, parity is a common optimum for matching and prediction.

3 INFORMATION LOSS AND PREDICTOR ERROR

For a predictor with normalized conditional distribution $q(\cdot \mid Z, V)$, excess log loss has two parts:

$$L(q, Z) - R^* = \underbrace{R(Z) - R^*}_{\text{code regret } \Delta(Z)} + \underbrace{\mathbb{E}_{Z,V}\, \mathrm{KL}\big(P_{X|Z,V} \,\|\, q(\cdot \mid Z, V)\big)}_{\text{predictor error}}.$$

Here $L(q, Z)$ is the predictor's expected log loss. Changing the encoder can change what the code retains; refitting a predictor to a frozen code can reduce only the second term.

We train 128-bit codes for 128-bit source strings at $p = 3/4$ and mask retention $a = 1/2$, using hard bits in the forward pass and a straight-through gradient for training (Bengio et al., 2013). The encoder receives the raw strings and their coordinatewise parity and starts by storing U_1 or $U_1 \oplus U_2$. This tests whether training preserves supplied useful information; it does not test discovery of parity.

The 36 encoder–predictor pairs cross three availabilities $r \in \{.5, .9, 1\}$, two starts, matching or direct prediction, and three seeds. Each encoder is trained with a product-Bernoulli predictor, its *co-trained predictor*. Matching uses that predictor's log probabilities to score 1024 candidates; direct prediction minimizes its log loss. Fresh product-Bernoulli and autoregressive predictors are then fitted to each frozen matching code. For each matching encoder, these predictors read its same stored 128 bits (Appendix B).

Simultaneous 95% bounds on the co-trained critics' matching gains and Proposition 2 establish positive regret for every predictor of nine matching codes: all three seeds of the raw start at $r = .5$ and both starts at $r = .9$ (Appendix B.4). These attained gains also exceed the initial codes' best matching gains, which are at most $.75 \log K$ for the raw code and $r \log K$ for parity. From the prediction-optimal starts (raw at $r = .5$ and parity at $r = .9$), training therefore improves matching while losing predictive information.

Predictive loss that refitting cannot remove. We bound the matching code's minimum risk from below by evaluating predictors without the other string. Target entropy minus their log loss lower-bounds information retained about each string separately. This information uses the same code budget as the prediction-optimal allocation, allowing the storage constraint to bound regret under the original workload.

Designed after the primary experiment, this analysis evaluates fixed models on fresh, independent draws of the source strings and masks. We mix each predictive distribution with 1% uniform mass over the complete masked target, making log loss bounded. Lower bounds on each matching code's minimum risk and upper bounds on the smoothed direct predictor's risk hold together with 95% confidence for these fitted models (Appendix C).

Figure 2(a,b) shows the two prediction-optimal starts at $r < 1$: raw at $r = .5$ and parity at $r = .9$. In all six pairs (three seeds per condition), the smoothed direct predictor outperforms *every* predictor of the matching code. Panels (c,d) show all eighteen comparisons; the other four groups are inconclusive, including both starts at $r = 1$, where parity is a common optimum. These bounds use a separate simultaneous guarantee from the matching-score analysis.

Predictor fitting and initialization. Refitting reduces much of the excess loss in both highlighted conditions. The positive lower bounds in Figure 2(a,b) show excess loss that no predictor of these codes can remove. The best evaluated smoothed predictors still lie well above these bounds, so the remaining excess can include both additional code regret and predictor error. Appendices C and B.4 give all bounds and the unsmoothed comparisons.

Initialization matters for both objectives. At $r = 1$, parity-start matching nearly attains the common prediction optimum (unsmoothed excess log loss .00039), while the raw start performs poorly. Direct prediction likewise remains worse from the nonoptimal starts at $r = .5$ and .9.

Initially, parity reveals neither string alone, but knowing either string and the code determines the other. At $r = .9$, all three parity-start matching codes instead retain at least 20.12 bits in total about the individual strings. The storage bound therefore implies less conditional mutual information between the strings than under the initial parity code (Appendix C.4). These bounds do not identify specific retained coordinates.

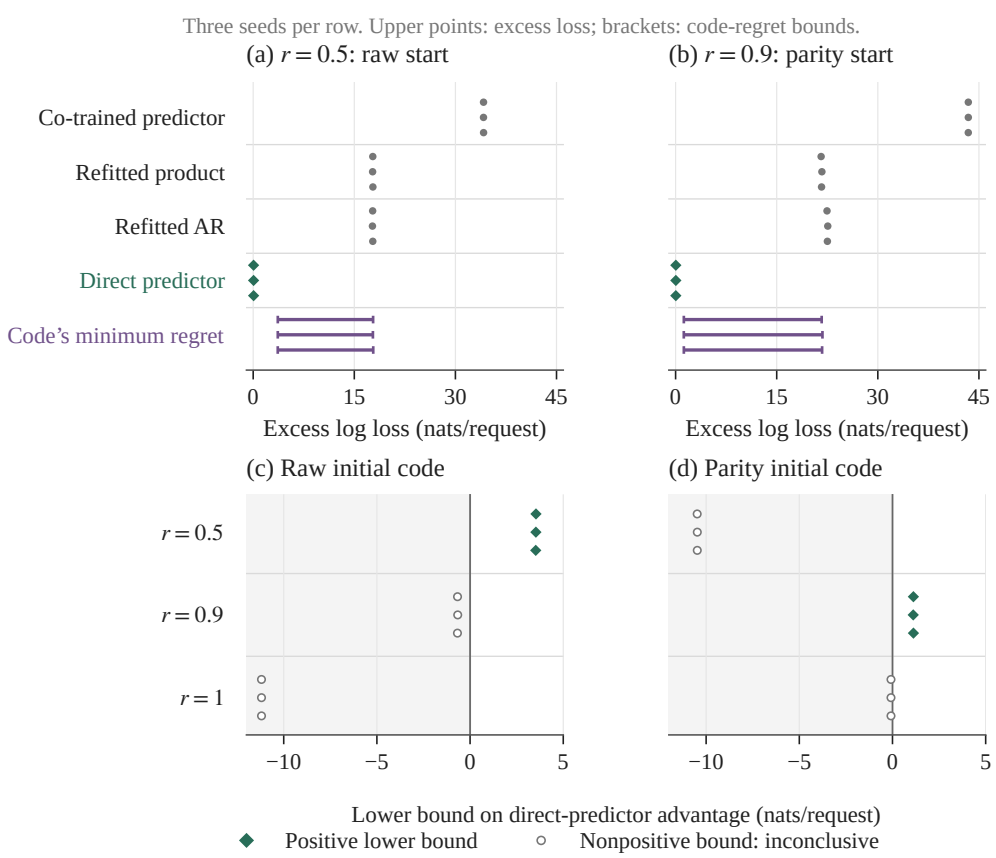

Figure 2: **Refitting reduces predictor error, but code regret remains.** (a,b) Excess log loss over R^* for the co-trained product-Bernoulli predictor and the refitted product-Bernoulli and autoregressive (AR) predictors on the same matching code, and the direct predictor on its separately trained code. Each row shows estimates for three fixed seeds. Brackets bound each matching code's regret using retained information (lower) and its best of three evaluated smoothed predictors (upper), not variation across seeds. Losses use a 1% uniform mixture over the complete masked target. (c,d) All eighteen paired comparisons. Positive lower bounds show that the smoothed direct predictor beats every predictor of the matching code. Nonpositive bounds are inconclusive. The two positive groups have minimum advantages 3.524 and 1.113 nats per request. All bounds hold simultaneously with 95% confidence for these fixed models.

Withholding the other string reverses the order. The same three parity-start pairs, trained at $r = .9$, also admit a comparison when the other string never arrives. A normalized predictor constructed from the already-evaluated matching-code predictors then beats *every* predictor of the paired direct code by at least 4.540 nats per request. At $r = .9$, the smoothed direct predictor beats every predictor of the matching code by at least 1.113 nats, under the same simultaneous bounds (Appendix C.4). The frozen codes' minimum risks are thus ordered oppositely: matching has retained information useful without the other string. This post-hoc analysis compares two availabilities; the raw-start pairs remain inconclusive.

We check sensitivity to the straight-through gradient by retraining with the derivative of a map that divides each preactivation row by its root mean square. The hard-bit forward map is unchanged; the original encoders serve as controls. Both direct-predictor advantages and the availability reversal persist across all three seeds under new simultaneous 95% bounds. The other four groups remain inconclusive (Appendix C.5).

4 Optimal allocation in a Gaussian model

The bit-string results bound predictive regret without finding the best matching code. We now solve matching over all linear subspaces of a Gaussian source. Here the later context is only a request; no complementary source observation is supplied.

Let $Y^{(j)} \sim \mathcal{N}(0, I_{n_j})$ be independent Gaussian source blocks, collected in Y. An independent request chooses block j with probability $p_j > 0$, and its target has coordinates

$$X_\ell = \sqrt{a_{j\ell}} Y_\ell^{(j)} + \sqrt{1 - a_{j\ell}} N_\ell, \qquad 1 > a_{j1} \geq \cdots \geq a_{jn_j} > 0,$$

with independent standard Gaussian noise. The quality $a_{j\ell}$ is the squared correlation between a source coordinate and its target. A linear code QY is formed before the request, with rank at most $R \leq \sum_j n_j$.

Retaining coordinate ℓ of block j provides $m_{j\ell} = -\frac{1}{2}\log(1 - a_{j\ell})$ nats about its target coordinate. Prediction therefore values that stored direction by $p_j m_{j\ell}$.

For this analysis the critic observes one target and K candidate codes. One uniformly placed code is drawn jointly with the target; the other codes are independent draws from $P(Z \mid J) = P(Z)$, since J is independent of Y. Let $B_j(k)$ be the optimal matching gain for request j when the code retains that block's k strongest directions. It equals $\log K$ minus the minimum expected candidate-index log loss. Set $B_j(0) = 0$. Appendix D specifies this Gaussian channel.

We include a predictive-information term with weight $\lambda \geq 0$; $\lambda = 0$ gives pure matching.

Theorem 3 (Exact finite allocation)**.** *For $K \geq 2$ and any finite predictive weight $\lambda \geq 0$, every linear code maximizing the Bayes matching gain plus $\lambda I(X; Z \mid V)$ under the rank budget has rank R. Its rowspace is a direct sum of leading signal subspaces within blocks. The numbers of directions retained in the blocks are obtained by taking the R largest weighted marginal gains*

$$p_j\{B_j(k) - B_j(k-1) + \lambda m_{jk}\}.$$

Conversely, every such selection is optimal. The gains strictly decrease within a block; ties give the corresponding freedom in the optimal code.

The proof expresses the objective through squared canonical correlations and bounds each projection's value by a fractional allocation of its rank among source directions. Strict convexity along pairwise transfers forces every relaxed optimum to use whole directions. Blockwise codes attain the bound, and decreasing marginal gains make greedy selection exact (Appendix D). The argument requires positive noise and averaging over independent Gaussian candidate sets.

We test whether training reaches the allocation in Figure 3 without restricting the subspace. Four fits cross $p = .75$ and $.10$ with population and sampled-candidate gradients, using the known-source Bayes critic and one common initialization.

At $p = .75$, every linear code with population matching loss at most $.0014$ nats per candidate set incurs more than 17.88 nats of predictive regret per request. Being within $.0004$ nats of the optimal matching loss suffices to meet this threshold (Appendix G). The population-gradient fit reaches the 14/50 allocation numerically after 8,192 updates; its independently evaluated 95% upper bound on matching loss, $.000822$, is below $.0014$. The sampled-candidate bound, $.001453$, is not. At $p = .10$, both matching and prediction improve from the common start. Appendix G.5 plots all four trajectories from their common start and gives simultaneous bounds for the four final codes (Figure 4). These fits demonstrate the objective's attainable preference, without establishing general training convergence.

How tightly matching is optimized matters. For two equal-quality n-dimensional blocks and rank at most n, hold unequal request probabilities, quality and candidate count fixed as n grows. Every exact matching optimum splits its directions equally to leading order, with bounded offsets that can depend on request frequency. Once both requests are easy to match, substantially different allocations can have almost the same matching loss while retaining different predictive information. Among codes within a fixed positive tolerance of the population matching optimum, the least predictive regret stays bounded. This least regret grows like $\log n$ for polynomially small tolerance and proportionally to n for exponentially small tolerance. The bounds characterize the best prediction achievable within each tolerance over all subspaces, rather than a training trajectory (Appendix E).

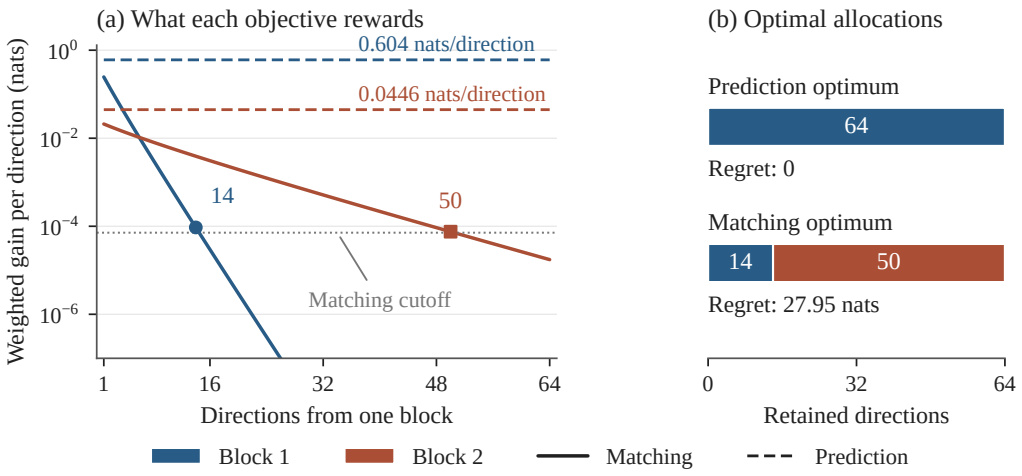

Figure 3: **Prediction and matching value stored directions differently.** Two 64-dimensional Gaussian blocks have qualities $(.8, .3)$, request probabilities $(.75, .25)$, rank budget 64 and two candidates. (a) Prediction's weighted gains are constant; matching gains diminish (logarithmic scale). (b) Prediction retains all 64 directions from block 1; matching selects 14 from block 1 and 50 from block 2, losing 27.95 nats of predictive information per request. Marked points are each block's last selected direction; the cutoff separates the 64th and 65th largest gains. Numerical population values illustrate Theorem 3, which covers all linear subspaces. Gains below the plotted range (10^{-7}) still enter the allocation.

For two blocks with equal dimension and quality but unequal request probabilities, a finite predictive weight aligns the optima. Strictly above an explicit threshold, excess in the combined population objective bounds predictive regret (Appendix D.2). Averaging only the predictive loss over coordinates divides its effective weight by the target dimension. With a learned critic and an analytically fitted Gaussian predictor on such a source, adding coordinate-averaged predictive log loss reduces code regret by 16–18% in two paired seeds. Summing that loss instead reduces regret by about 95%, at substantial matching cost (Appendix G.6).

When Gaussian objectives agree. The discrete request makes target blocks compete for the code. If instead (X, Y, V) is jointly Gaussian with positive-definite covariance, conditioning on V leaves a context-independent residual covariance. Canonical correlation analysis (Hotelling, 1936) then selects the same optimal subspaces for prediction and finite matching. The encoder still sees only Y; predictor and critic subtract the code's conditional mean after V arrives (Appendix F). The same subspaces optimize proper binary testing of conditional pairs, but near-optimal matching need not imply near-optimal prediction (Appendices F.1 and F.2).

5 Beyond the finite-candidate limit

Removing the candidate ceiling leaves the choice of divergence. The KL divergence from $P(X, Z \mid V)$ to $P(X \mid V)P(Z \mid V)$, averaged over V, equals $I(X; Z \mid V)$ and agrees with prediction. We study whether other divergences without the finite-candidate ceiling preserve this agreement under the same fixed-bit budget. Empirical comparisons find that the best divergence varies with the task (Lu et al., 2023).

We return to the binary source, changing only the coordinate-retention probabilities to request-dependent $a_1, a_2 \in (0, 1]$.

For a code that stores one complete string, prediction values the average number of requested bits it reveals. A power divergence instead averages an exponential of that count, giving more weight to masks that reveal many bits. For order $\alpha > 1$, define the pooled power divergence (van Erven & Harremoës, 2014)

$$\mathcal{D}_\alpha(Z) = \frac{\mathbb{E}_{V,Z,X}\left[\left\{\frac{P(Z,X|V)}{P(Z|V)P(X|V)}\right\}^{\alpha-1}\right] - 1}{\alpha - 1}.$$

The expectation averages over contexts V and, conditional on each context, over the joint distribution of (Z, X); zero-probability cells are omitted. At $\alpha = 1$, the limiting criterion is the predictive information $I(X; Z \mid V)$; at $\alpha = 2$, it is Pearson's χ^2 divergence.

Pearson divergence also underlies quadratic contrastive scores; spectral contrastive learning restricts these to shared inner products (HaoChen et al., 2021). Here we analyze the full conditional divergence. The moments are pooled across the prediction contexts; averaging their logarithms would define a different objective.

With $(p_1, p_2) = (p, 1-p)$, prediction assigns U_1, U_2 and $U_1 \oplus U_2$ information values proportional to $p_1 a_1$, $p_2 a_2$ and $r(p_1 a_1 + p_2 a_2)$, respectively. For the power criterion, the corresponding values of $(\alpha - 1)\mathcal{D}_\alpha$ are w_1, w_2 and $r(w_1 + w_2)$, where

$$w_j = p_j\big[(1 - a_j + a_j 2^{\alpha-1})^n - 1\big]. \tag{1}$$

A less frequent but more reliably observed string can therefore become more valuable to the power objective.

Raw and parity codes devote their information to individual strings or to dependence between them. Every randomized encoder obeys the same budget:

$$I(U_1; Z) + I(U_2; Z) + I(U_1; U_2 \mid Z) = I(U_1, U_2; Z) \le n \log 2.$$

Appendix H.2 uses this decomposition to prove $(\alpha - 1)\mathcal{D}_\alpha(Z) \le \max\{w_1, w_2, r(w_1 + w_2)\} + (w_1/a_1 + w_2/a_2)/(en \log 2)$. The best raw or parity code attains the leading term. The gap between a uniquely largest weight and the others controls a near-optimal code's departure from that allocation.

Agreement and disagreement under the same budget. When $a_1 = a_2$, the power weights and prediction weights rank the three allocations in the same order. Away from a tie, *every exact power optimum* has predictive regret bounded independently of string length, uniformly over $\alpha > 1$. This is a bounded-regret result, not equality of the finite-size optimizer sets. Approximate optimization must resolve the actual gap between the power weights.

Unequal retention changes the answer. Take request probabilities $(.75, .25)$, retentions $(.5, .9)$, and availability $r = .9$. Prediction favors parity. Pearson's objective instead forces nearly complete recovery of the second string from the code alone, although it is requested less often. At $n = 128$, attaining at least the population score of simply storing U_2 forces about 28 nats of predictive regret per request, including at every exact optimum. The proof controls whole-string decoding error and gives a finite regret interval (Appendix H.3).

Approaching KL does not guarantee small regret when string length also grows. With $\alpha = 1 + c/n$, the weights in (1) converge to $p_j(2^{ca_j} - 1)$. For this source, the limiting power weights of parity and storing U_2 cross at $c_\star \simeq 11.83$. Away from this crossing, exact optima have regret bounded in n below it and proportional to n above it. For approximately optimal sequences with gap $o(n)$ in $\mathcal{D}_\alpha$, the corresponding conclusions are sublinear regret below the crossing and linear regret above it. The relevant scale is $n(\alpha - 1)$: a smaller order can still accompany greater regret for a longer string (Table 1).

When the other string always arrives ($r = 1$), parity optimizes prediction and every power criterion with $\alpha > 1$. As availability approaches one, Pearson's recovery bound needs an objective tolerance resolving the rare requests without the other string (Appendix H).

Table 1: **Uncapped objectives need not preserve predictive information.** Evaluated bounds on every exact optimal code for the source with request probabilities $(.75, .25)$, retentions $(.5, .9)$ and availability $r = .9$. Entries are predictive regret in nats per request, not sampling intervals. Prediction-optimal losses are 5.32 and 21.29 nats at $n = 128$ and 512, respectively.

String length n	Power order α	$n(\alpha - 1)$	Predictive regret bounds
128	1 (prediction)	0	0
128	1.015625	2	$[0,\ 1.503]$
128	1.125	16	$[25.848,\ 29.614]$
512	1.03125	16	$[109.760,\ 113.403]$

A squared-error study jointly learns the code and target scales, with target covariance normalized to identity across requests and masks. The scales can move variance between requests; per-request normalization prevents this. At $r = 0$, the studied target family's population-optimal code stores the less frequent, more reliably observed string. From the prediction-optimal U_1 start, pooled normalization has over five nats/request more predictive regret than per-request normalization and direct prediction in both seeds; this separation is not established from U_2 (Appendix I).

6 Agreement cases and natural-data comparisons

Exact agreement for the binary source. Following conditional contrastive decomposition (Sordoni et al., 2021), match each selected bit separately given the request, mask, supplied other string and preceding selected bits. Use one independent fair negative and a uniformly random positive position. Normalizing a shared per-candidate score's exponentials over the two bit values defines a Bernoulli predictor q, with

$$L_{\mathrm{match}}(q, Z) = \tfrac{1}{2}\mathbb{E}|S| \log 2 + \tfrac{1}{2} L_{\mathrm{prediction}}(q, Z).$$

Candidates coincide half the time; otherwise their matching loss equals predictive log loss. Hence the losses have identical minimizers over any shared encoder–predictor class, without requiring successful fitting. This agreement uses m negatives and $2m$ scores for an m-bit request, with the stated prefixes; the encoder receives no later information (Appendix J).

Natural-data comparisons. On DeepFashion (Liu et al., 2016), predicting 1,000 annotations from 128-dimensional codes, refitting predictors to both frozen codes removes about 99.4% of the gap in summed binary cross-entropy between the co-trained matching and direct predictors. The remaining gap is statistically unresolved. On OCR-VQA (Mishra et al., 2019), two seeds disagree on the remaining gap, and reducing the code from 512 to 128 bits does not amplify it.

For small categorical codes learned from frozen image features, we compute minimum prediction loss exactly on the finite evaluation distribution. None of 42 paired comparisons on DeepFashion annotations and CUB bird attributes (Wah et al., 2011) shows a fitted direct predictor beating every predictor of the matching code. The comparisons reuse fits and evaluation data across selected and final checkpoints and are dependent. Where matching loses predictive information, the direct predictor's fitting error exceeds that loss; matching wins on both objectives in 22 pairs.

A separate one-bit DeepFashion quantizer gives a direct predictor fitted on training images an advantage of about $.006718$ nats/request over every predictor of the matching code on the finite evaluation distribution. The population matching order remains unresolved (Appendix K). These studies do not establish a natural-population information penalty.

7 Conclusion

Training objectives value retained information differently. For the binary and Gaussian sources studied here, we characterize optimal allocations and bound predictive regret from attained scores. The experiments separate information loss from predictor error; the agreement cases explain when the objectives align. How often these losses occur in natural representations remains open.

AI USE STATEMENT

We developed this work with assistance from generative AI tools, including OpenAI's GPT models. In iterative discussions, we raised conceptual questions, explored connections to the literature, suggested experiments and critically examined the emerging ideas. The tools helped formulate theoretical models, hypotheses and mathematical claims, and supplied derivations and proof arguments, including complete arguments for some results. This included developing the curvature argument for the Gaussian matching objective that yields an exact allocation rule over all linear subspaces in Section 4. We also used AI tools to find and understand related theoretical work and to design and implement experiments, including code for data preprocessing and synthetic-data generation. They assisted with analysis and interpretation of results, figure preparation, and refining the manuscript.

We reviewed the proofs, worked through additional derivations and checked relevant claims against primary sources. We also recomputed analytic examples and checked the consistency of the numerical results. AI tools assisted with these checks. We retain responsibility for the final content and its verification.

REPRODUCIBILITY STATEMENT

The appendices give the proofs, experimental protocols and checkpoint-selection rules. The supplementary code includes the original implementation and records for all 36 controlled encoder fits and all 18 paired comparisons. It provides CPU replay from saved states for two positive paired comparisons and a nonpositive control, and recomputes all 18 margins from saved statistics (`supplement/context_code_replay/`). This checks evaluation without retraining. Standard-library scripts in `numerical_support/` recompute the availability reversal and every bound of the straight-through-gradient comparison from saved sufficient statistics. That comparison's training source, numerical records and selected final states are supplied. CPU replay covers eight encoders and their 16 predictors, including positive, inconclusive and agreement controls, using the first 16 saved predictive rows per encoder. Portable binary reconstruction commands were checked with short training and resumption runs; the paper-scale fits and full evaluation banks were not repeated for this release. The `supplement/categorical_saved_moments/` package reconstructs finite-distribution objectives and all 42 pair margins from saved code–label distributions and fitted tables for 120 DeepFashion/CUB states, without encoder evaluation or bootstrap. All five figures can be regenerated from fixed inputs. The supplement includes setup instructions, a result-to-command map, tested environments, checksums and pinned image-feature provenance; it documents the remaining limits on retraining and raw-data reconstruction.

REFERENCES

Sanjeev Arora, Hrishikesh Khandeparkar, Mikhail Khodak, Orestis Plevrakis, and Nikunj Saunshi. A Theoretical Analysis of Contrastive Unsupervised Representation Learning. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5628–5637. PMLR, 2019. URL https://proceedings.mlr.press/v97/saunshi19a.html.

Parikshit Bansal, Ali Kavis, and Sujay Sanghavi. Understanding contrastive learning via Gaussian mixture models. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-3703. URL https://papers.nips.cc/paper_files/paper/2025/hash/a0bc6dfb643fd8b19b19f1f0e07c72c6-Abstract-Conference.html.

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013. URL https://arxiv.org/abs/1308.3432.

David Blackwell. Comparison of Experiments. In *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, pp. 93–102. University of California Press, 1951. URL https://digicoll.lib.berkeley.edu/record/112749.

Maurits Bleeker, Andrew Yates, and Maarten de Rijke. Reducing predictive feature suppression in resource-constrained contrastive image-caption retrieval. *Transactions on Machine Learning Research*, 2023. URL https://openreview.net/forum?id=T1XtOqrVKn.

Daniel Bolya, Po-Yao Huang, Peize Sun, Jang Hyun Cho, Andrea Madotto, Chen Wei, Tengyu Ma, Jiale Zhi, Jathushan Rajasegaran, Hanoona Rasheed, Junke Wang, Marco Monteiro, Hu Xu, Shiyu Dong, Nikhila Ravi, Daniel Li, Piotr Dollár, and Christoph Feichtenhofer. Perception encoder: The best visual embeddings are not at the output of the network. In *Advances in Neural Information Processing Systems*, volume 38, pp. 60884–60937, 2025. doi: 10.52202/085713-2036. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/57bc0a850255e2041341bf74c7e2b9fa-Paper-Conference.pdf.

Adrian Bulat, Yassine Ouali, and Georgios Tzimiropoulos. Compress & cache: Vision token compression for efficient generation and retrieval. In *Advances in Neural Information Processing Systems*, volume 38, pp. 31943–31968, 2025. doi: 10.52202/085713-1073. URL https://papers.neurips.cc/paper_files/paper/2025/hash/2e01083b381b4865919b4915ef32e3d2-Abstract-Conference.html.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A Simple Framework for Contrastive Learning of Visual Representations. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 1597–1607. PMLR, 2020. URL https://proceedings.mlr.press/v119/chen20j.html.

Ting Chen, Calvin Luo, and Lala Li. Intriguing properties of contrastive losses. In *Advances in Neural Information Processing Systems*, volume 34, pp. 11834–11845, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/628f16b29939d1b060af49f66ae0f7f8-Abstract.html.

Yann Dubois, Benjamin Bloem-Reddy, Karen Ullrich, and Chris J. Maddison. Lossy compression for lossless prediction. In *Advances in Neural Information Processing Systems*, volume 34, pp. 14014–14028, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/7535bbb91c8fde347ad861f293126633-Abstract.html.

John C. Duchi, Khashayar Khosravi, and Feng Ruan. Multiclass classification, information, divergence and surrogate risk. *The Annals of Statistics*, 46(6B):3246–3275, 2018. doi: 10.1214/17-AOS1657. URL https://arxiv.org/abs/1603.00126.

Alex Dytso, H. Vincent Poor, and Shlomo Shamai (Shitz). Conditional mean estimation in Gaussian noise: A meta derivative identity with applications. *IEEE Transactions on Information Theory*, 69 (3):1883–1898, 2023. doi: 10.1109/TIT.2022.3216012. URL https://par.nsf.gov/servlets/purl/10444847.

Adam Foster, Martin Jankowiak, Matthew O'Meara, Yee Whye Teh, and Tom Rainforth. A unified stochastic gradient approach to designing Bayesian-optimal experiments. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pp. 2959–2969. PMLR, 2020. URL https://proceedings.mlr.press/v108/foster20a.html.

Robert G. Gallager. A simple derivation of the coding theorem and some applications. *IEEE Transactions on Information Theory*, 11(1):3–18, 1965. doi: 10.1109/TIT.1965.1053730. URL https://web.mit.edu/6.441/www/reading/IT-V11-N1.pdf.

Mathieu Germain, Karol Gregor, Iain Murray, and Hugo Larochelle. MADE: Masked autoencoder for distribution estimation. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 881–889, 2015. URL https://proceedings.mlr.press/v37/germain15.html.

Tilmann Gneiting and Adrian E. Raftery. Strictly Proper Scoring Rules, Prediction, and Estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007. doi: 10.1198/016214506000001437. URL https://sites.stat.washington.edu/people/raftery/Research/PDF/Gneiting2007jasa.pdf.

Dongning Guo, Shlomo Shamai, and Sergio Verdú. Mutual information and minimum mean-square error in Gaussian channels. *IEEE Transactions on Information Theory*, 51(4):1261–1282, 2005. doi: 10.1109/TIT.2005.844072. URL https://arxiv.org/abs/cs/0412108.

Dongning Guo, Yihong Wu, Shlomo Shamai (Shitz), and Sergio Verdú. Estimation in Gaussian noise: Properties of the minimum mean-square error. *IEEE Transactions on Information Theory*, 57(4): 2371–2385, 2011. doi: 10.1109/TIT.2011.2111010. URL https://doi.org/10.1109/TIT.2011.2111010.

Jeff Z. HaoChen, Colin Wei, Adrien Gaidon, and Tengyu Ma. Provable guarantees for self-supervised deep learning with spectral contrastive loss. In *Advances in Neural Information Processing Systems*, volume 34, pp. 5000–5011, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/27debb435021eb68b3965290b5e24c49-Abstract.html.

Martin E. Hellman and Josef Raviv. Probability of error, equivocation, and the Chernoff bound. *IEEE Transactions on Information Theory*, 16(4):368–372, 1970. doi: 10.1109/TIT.1970.1054466. URL https://www-ee.stanford.edu/~hellman/publications/07.pdf.

Dan Hendrycks and Kevin Gimpel. Gaussian Error Linear Units (GELUs), 2016. URL https://arxiv.org/abs/1606.08415.

Wassily Hoeffding. Probability Inequalities for Sums of Bounded Random Variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963. doi: 10.1080/01621459.1963.10500830.

Harold Hotelling. Relations between two sets of variates. *Biometrika*, 28(3–4):321–377, 1936. doi: 10.1093/biomet/28.3-4.321. URL https://doi.org/10.1093/biomet/28.3-4.321.

Diederik P. Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. URL https://arxiv.org/abs/1412.6980.

Shigeaki Kuzuoka, Akisato Kimura, and Tomohiko Uyematsu. Universal coding for lossless and lossy complementary delivery problems. *arXiv preprint arXiv:0802.1567*, 2008. URL https://arxiv.org/abs/0802.1567.

Kyungmin Lee and Jinwoo Shin. RényiCL: Contrastive representation learning with skew Rényi divergence. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 6463–6477. Curran Associates, Inc., 2022. doi: 10.52202/068431-0468. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/2a7157c84dcf263f77b37d6c11d7d149-Paper-Conference.pdf.

Licong Lin and Song Mei. A statistical theory of contrastive learning via approximate sufficient statistics. In *Advances in Neural Information Processing Systems*, volume 38, pp. 127952–127996, 2025. doi: 10.52202/085713-4262. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/b9e88ae0308cf82d0b0f634ddbdf809a-Abstract-Conference.html.

Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. DeepFashion: Powering robust clothes recognition and retrieval with rich annotations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1096–1104, 2016. doi: 10.1109/CVPR.2016.124. URL https://www.cv-foundation.org/openaccess/content_cvpr_2016/html/Liu_DeepFashion_Powering_Robust_CVPR_2016_paper.html.

Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*, 2019. URL https://arxiv.org/abs/1711.05101.

Yiwei Lu, Guojun Zhang, Sun Sun, Hongyu Guo, and Yaoliang Yu. f-MICL: Understanding and generalizing InfoNCE-based contrastive learning. *Transactions on Machine Learning Research*, 2023. URL https://openreview.net/forum?id=ZD03VUZmRx.

Zhe Ma, Jianfeng Dong, Zhongzi Long, Yao Zhang, Yuan He, Hui Xue, and Shouling Ji. Fine-Grained Fashion Similarity Learning by Attribute-Specific Embedding Network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 11741–11748, 2020. doi: 10.1609/aaai.v34i07.6845.

Andreas Maurer and Massimiliano Pontil. Empirical Bernstein bounds and sample variance penalization. *arXiv preprint arXiv:0907.3740*, 2009. URL https://arxiv.org/abs/0907.3740.

Anand Mishra, Shashank Shekhar, Ajeet Kumar Singh, and Anirban Chakraborty. OCR-VQA: Visual question answering by reading text in images. In *International Conference on Document Analysis and Recognition*, pp. 947–952, 2019. doi: 10.1109/ICDAR.2019.00156. URL https://ocr-vqa.github.io/.

XuanLong Nguyen, Martin J. Wainwright, and Michael I. Jordan. On surrogate loss functions and f-divergences. *The Annals of Statistics*, 37(2):876–904, 2009. doi: 10.1214/08-AOS595. URL https://people.eecs.berkeley.edu/~jordan/papers/nguyen-wainwright-jordan-aos-2009.pdf.

Miquel Payaró and Daniel P. Palomar. Hessian and concavity of mutual information, differential entropy, and entropy power in linear vector Gaussian channels. *IEEE Transactions on Information Theory*, 55(8):3613–3628, 2009. doi: 10.1109/TIT.2009.2023749. URL https://palomar.home.ece.ust.hk/papers/2009/PayaroPalomar_TransIT2009_Hessian.pdf.

Ben Poole, Sherjil Ozair, Aäron van den Oord, Alexander A. Alemi, and George Tucker. On variational bounds of mutual information. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5171–5180. PMLR, 2019. doi: 10.48550/arXiv.1905.06922. URL https://proceedings.mlr.press/v97/poole19a.html.

András Prékopa. On logarithmic concave measures and functions. *Acta Scientiarum Mathematicarum*, 34:335–343, 1973. URL https://acta.bibl.u-szeged.hu/14411/1/math_034_335-343.pdf.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pp. 8748–8763. PMLR, 2021. URL https://proceedings.mlr.press/v139/radford21a.html.

Joshua Robinson, Li Sun, Ke Yu, Kayhan Batmanghelich, Stefanie Jegelka, and Suvrit Sra. Can contrastive learning avoid shortcut solutions? In *Advances in Neural Information Processing Systems*, volume 34, pp. 4974–4986, 2021. URL https://proceedings.neurips.cc/paper/2021/file/27934a1f19d678a1377c257b9a780e80-Paper.pdf.

Nikunj Saunshi, Jordan Ash, Surbhi Goel, Dipendra Misra, Cyril Zhang, Sanjeev Arora, Sham Kakade, and Akshay Krishnamurthy. Understanding Contrastive Learning Requires Incorporating Inductive Biases. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 19250–19286. PMLR, 2022. URL https://proceedings.mlr.press/v162/saunshi22a.html.

Alessandro Sordoni, Nouha Dziri, Hannes Schulz, Geoff Gordon, Philip Bachman, and Remi Tachet Des Combes. Decomposed mutual information estimation for contrastive representation learning. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 9859–9869. PMLR, 2021. doi: 10.48550/arXiv.2106.13401. URL https://proceedings.mlr.press/v139/sordoni21a.html.

Jiyuan Tao, Juyoung Jeong, and M. Seetharama Gowda. Some log and weak majorization inequalities in Euclidean Jordan algebras. *Linear and Multilinear Algebra*, 70(16):3189–3206, 2022. doi: 10.1080/03081087.2020.1830020. URL https://arxiv.org/abs/2003.12377v2.

Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What Makes for Good Views for Contrastive Learning? In *Advances in Neural Information Processing Systems*, volume 33, pp. 6827–6839, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/4c2e5eaae9152079b9e95845750bb9ab-Abstract.html.

Yao-Hung Hubert Tsai, Martin Q. Ma, Muqiao Yang, Han Zhao, Louis-Philippe Morency, and Ruslan Salakhutdinov. Self-supervised representation learning with relative predictive coding. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=068E_JSq9O.

Yao-Hung Hubert Tsai, Tianqin Li, Martin Q. Ma, Han Zhao, Kun Zhang, Louis-Philippe Morency, and Ruslan Salakhutdinov. Conditional contrastive learning with kernel. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=AAJLBoGt0XM.

Michael Tschannen, Josip Djolonga, Paul K. Rubenstein, Sylvain Gelly, and Mario Lucic. On Mutual Information Maximization for Representation Learning. In *International Conference on Learning Representations*, 2020. URL https://arxiv.org/abs/1907.13625.

Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, Olivier Hénaff, Jeremiah Harmsen, Andreas Steiner, and Xiaohua Zhai. SigLIP 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features, 2025. URL https://arxiv.org/abs/2502.14786.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. URL https://arxiv.org/abs/1807.03748.

Tim van Erven and Peter Harremoës. Rényi divergence and Kullback–Leibler divergence. *IEEE Transactions on Information Theory*, 60(7):3797–3820, 2014. doi: 10.1109/TIT.2014.2320500. URL https://arxiv.org/abs/1206.2459v2.

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The Caltech-UCSD Birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. URL https://authors.library.caltech.edu/records/cvm3y-5hh21/files/CUB_200_2011.pdf.

Yihao Xue, Siddharth Joshi, Eric Gan, Pin-Yu Chen, and Baharan Mirzasoleiman. Which features are learnt by contrastive learning? On the role of simplicity bias in class collapse and feature suppression. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 38938–38970. PMLR, 2023. URL https://proceedings.mlr.press/v202/xue23d.html.

Yuling Yao, Aki Vehtari, Daniel Simpson, and Andrew Gelman. Using stacking to average Bayesian predictive distributions (with discussion). *Bayesian Analysis*, 13(3):917–1007, 2018. doi: 10.1214/17-BA1091. URL https://doi.org/10.1214/17-BA1091.

SUPPORTING MATERIAL

The appendices follow the progression from finite-bit codes to Gaussian allocation, other objectives and natural-data comparisons.

Table 2: Guide to the theoretical and empirical support.

Main result or comparison	Supporting material
Finite-bit optimum and attained-score bounds	Appendix A
Learned codes, predictor refits and availability reversal	Appendices B–C
Gaussian allocation, precision and training	Appendices D–G
Power divergences and learned targets	Appendices H–I
Exact binary agreement	Appendix J
Natural-data comparisons and predictor controls	Appendix K

The anonymous `numerical_support/appendix_records/README.md` indexes complete endpoint tables, settings and verification records.

A FINITE STORAGE WITH COMPLEMENTARY INFORMATION AT PREDICTION TIME

An arbitrary randomized encoder of independent uniform n-bit strings U_1, U_2 stores a code Z with at most 2^n values. A later request J chooses string 1 with probability $p > 1/2$; a mask S independently retains each requested coordinate with probability $a \in (0, 1]$; and $P(E = 1) = r$. These variables are mutually independent and independent of the source and encoder randomness. At $E = 1$ the complete other string is supplied, giving

$$V = (J, E, S,\ U_{3-J} \text{ when } E = 1), \qquad X = U_{J,S}.$$

The encoder does not see J, E, S; the supplied other string is outside the n-bit budget. Logarithms are natural; divide by $\log 2$ for bits.

A.1 THE PREDICTION OPTIMUM

Put $h = n \log 2$, $A = ah$, $D = I(U_1, U_2; Z)$, and $I_j = I(U_j; Z)$. Independence and the code budget give

$$I_1, I_2 \geq 0, \qquad I_1 + I_2 \leq D \leq H(Z) \leq h. \tag{2}$$

Write

$$M_j = I(U_{j,S}; Z \mid S), \qquad N_j = I(U_{j,S}; Z \mid S, U_{3-j}).$$

These are the actual masked information quantities, not simply rescaled unmasked information. The predictive information and Bayes log loss are

$$\begin{aligned} G(Z) &= (1 - r)\{pM_1 + (1 - p)M_2\} + r\{pN_1 + (1 - p)N_2\}, \\ R(Z) &= A - G(Z). \end{aligned} \tag{3}$$

Proposition 4 (Proposition 1, restated)**.** *For every finite n, the minimum prediction loss is*

$$R^* = A[1 - \max(p, r)].$$

Storing U_1 attains it when $r \leq p$, and storing the coordinatewise parity $U_1 \oplus U_2$ attains it when $r \geq p$.

Proof. The erasure inequality gives

$$M_j \leq aI_j, \qquad N_1 \leq a(D - I_2), \qquad N_2 \leq a(D - I_1).$$

To see the first inequality, expand $I(U_S; Z)$ in increasing coordinate order for each fixed subset S. Independence of the coordinates implies

$$I(U_i; Z \mid U_{S\cap[i-1]}) \le I(U_i; Z \mid U_{[i-1]}).$$

Their difference is a nonnegative conditional mutual information after conditioning on Z. Average the indicator $1\{i \in S\}$ to obtain $I(U_S; Z \mid S) \le aI(U; Z)$. The same argument holds at each supplied value of the independent opposite string. Finally, $I(U_1; Z \mid U_2) = D - I_2$ and likewise for string 2. It follows that

$$G(Z) \le a\{rD + (p-r)I_1 + (1-p-r)I_2\}.$$

The four vertices of the region in (2), after division by h, give values $0, p, 1-p, r$. Its maximum is $\max(p, r)$. The raw and parity codes attain this bound, including after masking: their information gains are pA and rA, respectively. □

A.2 A CORRECTLY CONDITIONED CANDIDATE COMPARISON

Here candidates are targets. At fixed $V = v$, draw an index uniformly from $[K]$, draw $X_1, \ldots, X_K$ independently from $P_{X|v}$, and draw Z from $P_{Z|X_I,v}$. Equivalently, draw one actual positive pair (X, Z), insert X into a uniform position, and draw all other targets independently from their conditional marginal. For this source that marginal is uniform on the requested masked bits even when the opposite string is supplied. All candidates share the same realized context.

A normalized critic $\widehat{p}(i \mid Z, V, X_{1:K})$ has population gain

$$f = \log K + \mathbb{E} \log \widehat{p}(I \mid Z, V, X_{1:K}), \qquad C = \log K.$$

A shared per-candidate scorer is permutation-equivariant, so its fixed positive position is equivalent in expectation to a uniform index; a position-aware classifier would not be. Appendix B gives the implemented donor sampling and independent final evaluation units.

The target-candidate and code-candidate experiments can have different values for a general encoder. The bounds below use their shared conditional information inequality, as in conditional InfoNCE (Sordoni et al., 2021, Proposition 1); they do not require equality of the two objectives.

Proposition 5 (Finite matching bounds predictive loss; general form of Proposition 2)**.** *Let $\Delta(Z) = R(Z) - R^*$. For any stochastic encoder satisfying the budget and any normalized critic in the stated experiment,*

$$\begin{aligned} f \le\ & p(1-r)\min(C, aI_1) + pr\min(C, a(D-I_2)) \\ & + (1-p)(1-r)\min(C, aI_2) + (1-p)r\min(C, a(D-I_1)). \end{aligned} \tag{4}$$

In particular, at $p = 3/4$,

$$r = 1/2: \qquad \Delta(Z) \ge 4[f - \tfrac{7}{8}C]_+, \tag{5}$$

$$r = 9/10: \qquad \Delta(Z) \ge \max\{0,\ 2(f - \tfrac{9}{10}C),\ 26f - \tfrac{126}{5}C\}. \tag{6}$$

These bounds concern the best possible predictor of the stored code. They make no assumption that training attains a global optimum.

Proof. At a fixed context, let $q_v = P_{Z|v}$. The candidate-index chain rule gives

$$\begin{aligned} I(I; Z, X_{1:K} \mid v) = I(X; Z \mid v) \\ - \mathbb{E}_{X_{1:K}} D_{\mathrm{KL}}\left(\frac{1}{K} \sum_i P_{Z|X_i,v} \,\middle\|\, q_v \right). \end{aligned}$$

The attained normalized log score is at most this index information, which is also at most C. Average over masks and supplied values, apply concavity of $t \mapsto \min(C, t)$, and then use the erasure inequalities from the preceding proof. This proves (4); no constancy of conditional information across supplied values is assumed.

At $r = 1/2$, the prediction bound can be written as

$$\Delta \ge a\{\tfrac{3}{4}(h - D) + \tfrac{1}{4}(D - I_1 - I_2) + \tfrac{1}{2}I_2\} \ge \tfrac{1}{2}aI_2.$$

Cap the other three modes in (4) to obtain $f \le \frac{7}{8}C + \frac{1}{8}\min(C, aI_2)$, proving (5).

At $r = 9/10$,

$$\Delta \ge a\{\tfrac{9}{10}(h - D) + \tfrac{3}{20}I_1 + \tfrac{13}{20}I_2\}.$$

Put $t_j = \min(C, aI_j)$. The two modes with the other string supplied contribute at most $\frac{9}{10}C$, so

$$f \le \tfrac{9}{10}C + \tfrac{3}{40}t_1 + \tfrac{1}{40}t_2, \qquad \Delta \ge \tfrac{3}{20}t_1 + \tfrac{13}{20}t_2.$$

Multiplying the first residual bound by 2 gives the term $2(f - \frac{9}{10}C)$ in (6). Multiplication by 26 gives $26(f - \frac{9}{10}C) \le \Delta + \frac{9}{5}t_1 \le \Delta + \frac{9}{5}C$, giving the term $26f - \frac{126}{5}C$. Nonnegative regret completes the result. □

If all complementary information arrives ($r = 1$), parity is a common prediction and finite-candidate matching optimum. Given the opposite string, it recovers the requested target exactly. Observing that target is maximally informative for identifying its position among independently drawn candidates. Candidate collisions can still leave index ambiguity, so the common optimum need not have zero matching loss, but there is no unavoidable conflict between the objectives at this boundary.

A.3 An eight-bit example

Set $n = 8$, $p = 3/4$, $r = 9/10$, $a = 1$, and $K = 2$. Consider three codes: all eight bits of U_1, all eight coordinatewise parities, and $(U_{1,1}, U_{2,2}, U_{1,3} \oplus U_{2,3}, \ldots, U_{1,8} \oplus U_{2,8})$. The last code uses disjoint coordinate positions. For each mode it reveals exactly t requested bits. A negative target agrees on all of them with probability 2^{-t}, so its Bayes matching loss is 2^{-t} bits. Averaging the four request/availability modes gives Table 3.

Table 3: **The eight-bit comparison.** Exact Bayes losses for the introduction's three codes. Prediction losses and regret are bits per request; matching losses are bits per two-candidate set.

Code	Prediction loss	Matching loss	Prediction regret
Raw string 1	2	$259/1024$	1.2
Parity	0.8	$53/512$	0
One raw bit per string, six parities	1.6	$73/1280$	0.8

The third code improves matching at a predictive cost. It is a feasible example, not asserted to be a global matching optimum. Its matching gain is $(1207/1280)\log 2$ nats. Substitution into (6) gives a lower bound $(11/128)\log 2$ on prediction regret for *every* code attaining at least this gain. That bound is smaller than this particular code's actual regret $0.8 \log 2$.

A.4 From independent candidate sets to a population statement

For a fixed fitted critic, replace its normalized probabilities by $\widetilde{p}_i = (1 - \tau)\widehat{p}_i + \tau/K$, where $\tau \in (0, 1)$. In this subsection, f_τ and $\overline{f}_\tau$ denote the population and empirical gains of this smoothed critic. Its per-set gain belongs to $[\log \tau, \log K]$. Construct each independent evaluation unit from one source pair and four mode scores, weighted by $p(1 - r), pr, (1 - p)(1 - r), (1 - p)r$. The four modes may share their source pair; they are not four independent samples. The weighted unit has the same range and the required population expectation. These units must be fresh relative to all fitting and selection of the encoder and critic.

For T independent complete units and M frozen critics, a union bound and Hoeffding's inequality (Hoeffding, 1963) give simultaneous probability at least $1 - \alpha$ that every attained gain is at least

$$\underline{f}_\tau = \overline{f}_\tau - \log(K/\tau)\sqrt{\frac{\log(M/\alpha)}{2T}}.$$

At $K = 1024$, $\tau = .01$, $T = 65536$, $M = 36$, and $\alpha = .05$, the subtracted radius is 0.0817358695 nats. This statement conditions on the frozen fitted encoders and critics. Substituting $\underline{f}_\tau$ in (5) or (6)

gives a lower bound on regret for every predictor of each code. Fitting a predictor afterward does not change that stored information. Selecting among additional critics using these same evaluation units would require a different confidence accounting or separate selection data.

B CONTROLLED NEURAL ENCODING EXPERIMENT

B.1 SOURCE, OBJECTIVES AND TRAINING CONDITIONS

We use Appendix A's source with $n = 128$, $p = .75$, $a = .5$, and $r \in \{.5, .9, 1\}$, represented as signs. The encoder stores exactly 128 hard bits, serialized little-endian into 16 bytes, before seeing the request, availability or mask. The supplied opposite string is outside that budget. The two requests and two availability outcomes give four modes per source pair, with weights $[.75(1-r), .75r, .25(1-r), .25r]$; together they form one independent evaluation unit.

We cross three availabilities, two initial codes (U_1 or parity), two objectives and three seeds: 36 encoders. Matching uses unsmoothed cross-entropy with $K = 1024$. Direct training uses the NLL of a normalized distribution over the complete masked target, summing over its selected bits. Each fit completes 8,192 AdamW updates (Loshchilov & Hutter, 2019) on batches of 1,024 source rows, and uses the final scheduled checkpoint without validation selection. AdamW uses moments $(.9, .999)$, offset 10^{-8}, weight decay 10^{-4}, and global gradient clipping at 5. At update $t = 0, \ldots, 8191$, the learning rate is $3 \times 10^{-4}[.05 + .95(1 + \cos(\pi t/8192))/2]$. Conditions within a seed share source strings, requests, masks and the uniform draw defining availability.

Each of the 18 matching encoders then receives fresh product-Bernoulli and autoregressive (AR) predictors, each fitted by NLL for 8,192 updates on a separate stream with the encoder frozen. Their fits neither select an encoder nor enter the original matching-score event. They use the same optimizer and rate schedule, with clipping restricted to predictor parameters. Initialization and stream-sharing details are in the `numerical_support/appendix_records/` methods record.

B.2 ENCODER AND NORMALIZED PREDICTORS

A seeded Gaussian QR decomposition, with triangular-diagonal signs fixing column signs, supplies a fixed 256×256 rotation. The encoder sums a bias-free linear skip from $[U_1, U_2, U_1 \odot U_2] \in \mathbb{R}^{384}$ to 128 preactivations and a 256–512–128 GELU (Hendrycks & Gimpel, 2016) residual on the rotated source. The residual's final layer starts at zero; the skip copies U_1 or parity. Both objectives share each seed/start initialization. Forward computation thresholds at zero, packs positive signs as one bits, and unpacks the bytes before reading. Only the straight-through gradient (Bengio et al., 2013) uses preactivations; no continuous or item-specific state bypasses the code.

For code z, other string u and availability b, put $c = [z, bu, z \odot bu, b] \in \mathbb{R}^{385}$. The product-Bernoulli predictor adds linear 385–256 and GELU 385–512–256 branches, yielding 128 logits per request. Its loss sums Bernoulli NLL over coordinates selected by the request mask. Co-trained logits initially have magnitude 12 for bits determined by the starting code and zero otherwise. Fresh product predictors use small random linear weights and a zero final residual layer.

The AR predictor has two 512-unit GELU layers, with separate context maps into both layers and the output. Context comprises c, two request indicators, and 128 mask indicators. MADE-style target connections (Germain et al., 2015) give hidden units degrees cycling through $0, \ldots, 127$, nondecreasing hidden masks, and strictly smaller-degree output masks. Target inputs are multiplied by the mask. Thus bit i sees only earlier revealed target values, and the product over revealed coordinates is normalized for every mask.

B.3 SAMPLING AND THE FIXED INFERENCE IMPLEMENTATION

For training anchor i, each other batch row supplies its full target string chosen by J_i, scored with anchor mask S_i. The product-density normalizer cancels in the candidate softmax. Final candidate sets instead draw 1,023 fresh complete uniform negatives per requested string and anchor before masking. These negatives are conditionally correct even with the opposite string supplied. A shared per-candidate scorer is permutation-equivariant, justifying the implemented positive position zero.

Training uses FP32. The weights after the final scheduled update are separately exported to FP64; every reported evaluation applies this fixed encoder to groups of 16 source rows, padding a short final chunk with all-positive rows and discarding padding outputs. This makes the evaluated codes independent of how inputs are batched; we do not claim exact real-arithmetic behavior or floating-point invariance on all inputs.

The FP32/FP64 sensitivity comparison found 12 differing bits across eight matching encoders, all near zero preactivation; independent replay reproduced them. The fixed FP64 implementation remains the evaluated encoder. The comparison counts and replay scope are in the numerical record.

Each of the 36 co-trained endpoints has 65,536 independent evaluation units, each containing all four modes. The original event uses a .01 uniform mixture of candidate probabilities only for bounded evaluation, not training. Every declared predictor also has 65,536 prediction rows on a fresh source-and-mask sampling stream, paired descriptively across predictors and using no candidates. Matching gains are nats per candidate set; prediction losses are nats per request. Mode covariance is retained within each complete row. The numerical record keeps every endpoint, mode score, conditional SE and sensitivity count; prediction panels neither select encoders nor replace matching gains.

B.4 Primary results and verification

The primary matching-score event follows Appendix A's confidence construction for 36 fixed co-trained critics. It bounds code regret, without upper-bounding the unsmoothed predictors' population risks; Appendix C supplies the separate direct-predictor comparison. `numerical_support/primary_endpoints.csv` gives the unsmoothed results for all 36 encoders and their 72 prediction panels, with conditional standard errors, and `numerical_support/appendix_records/context_condition_means.csv` the six condition means. Co-trained, product and AR predictors share the matching code; direct NLL uses a separately trained code from the same start. These empirical losses, in nats per request, and matching gains, in nats per candidate set, are distinct from the secondary population bounds for smoothed predictors.

The source optima are 11.090355, 4.436142 and zero nats for $r = .5, .9, 1$. With target entropy $H_0 = 64 \log 2$ nats, the codes storing U_1, parity, half of each string, or nothing have risks $H_0/4, (1 - r)H_0, H_0/2, H_0$, respectively. A full-source code has zero risk but costs 32 bytes, twice the budget. These analytic controls explain the primary starts; they are not learned fits. Nine matching endpoints have positive original lower bounds: the three raw starts at $r = .5$ (at least 2.328 nats), and the raw and parity starts at $r = .9$ (at least 1.028 and $.623$ nats, respectively, in each seed). A zero lower bound only means that this sufficient inequality is uninformative.

The same event shows that all nine codes improve matching over their initial codes, even with optimal initial critics. Requested U_2 bits are independent of the raw U_1 code, including when U_1 is supplied, so its gain is at most $p \log K$. Without the other string, parity is independent of the target, giving an overall cap $r \log K$. The attained-score bounds exceed these caps. From prediction-optimal starts (raw at $r = .5$, parity at $r = .9$), positive regret also proves a loss of predictive information during training.

The numerical record retains the successful $r = 1$ parity start and the poor $r = 1$ raw start. Direct prediction is also start-dependent: its parity start at $r = .5$ and raw start at $r = .9$ remain far from the source optimum. The study tests rewriting of supplied useful codes, not discovery of parity, global neural optimization or natural-data prevalence.

Verification scope. Saved-score reaggregation covered every endpoint and evaluation chunk; independent forward replay checked bounded prefixes and every saved set with FP32/FP64 differences. Exact coverage, conditional standard errors and reconstruction discrepancies are in the "Controlled matching experiment" sections of `numerical_support/appendix_records/METHODS_AND_CHECKS.md`. Training was not replayed; these checks do not establish exact-real behavior on every possible input.

C COMPARING A DIRECT PREDICTOR WITH EVERY PREDICTOR OF A MATCHING CODE

This analysis, designed after the primary experiment, compares the saved direct predictors with every predictor of their paired matching codes on fresh independent rows. Its simultaneous confidence bound covers these fixed models and selection among their evaluated predictors, without further fitting. It gives no guarantee over future fits.

C.1 FROM NORMALIZED PREDICTION TO RETAINED INFORMATION

For a realized subset S, the conditional target marginal is uniform, with probability mass $u = 2^{-|S|}$ at each target value. For any fitted normalized predictor q, mix its distribution over the complete masked target with this uniform distribution:

$$q^\tau = (1-\tau)q + \tau u, \qquad \tau = .01, \qquad g = \log \frac{q^\tau(X)}{u(X)}.$$

The mixture is applied to the full conditional probability of the revealed target, after multiplying its Bernoulli or autoregressive factors. It is not an independent mixture of each bit's probability. In terms of its original masked NLL ℓ, the observable is

$$g = \operatorname{logaddexp}\big(\log(.99) + |S|\log 2 - \ell,\ \log(.01)\big).$$

It lies in $[\log(.01), 128\log 2]$. Empty subsets give exactly zero; negative values are retained. Write g_{je} for g on requests for string j with $E = e$. Conditional KL nonnegativity gives, without the other string,

$$\mathbb{E}g_{j0} \le M_j, \qquad M_j = I(U_{j,S}; Z \mid S).$$

Thus an imperfect normalized predictor can demonstrate that the code retains information about a source string without the other. It need not recover all of it.

The information region from Appendix A also implies

$$r = .5: \quad \Delta(Z) \ge \tfrac{1}{2}M_2, \qquad r = .9: \quad \Delta(Z) \ge \tfrac{3}{20}M_1 + \tfrac{13}{20}M_2. \tag{7}$$

These follow from the two regret decompositions in the proof of Proposition 5, using $M_j \le aI_j$. The erasure factor is already included in M_j.

For each predictor form the complete weighted row

$$G = .75(1-r)g_{10} + .75rg_{11} + .25(1-r)g_{20} + .25rg_{21}.$$

The risk of the smoothed predictor is exactly $L(q^\tau, Z) = 64\log 2 - \mathbb{E}G$. A lower bound on $\mathbb{E}G$ therefore provides an upper risk bound for this actual predictor. It does not bound the original unsmoothed predictor's risk above: smoothing can improve adverse tails. All comparisons below name the smoothed direct predictor explicitly.

C.2 ONE FIXED SIMULTANEOUS EVALUATION

All 36 fixed encoders and 72 attached predictors receive 65,536 complete rows from one fresh source/mask stream paired across endpoints. The fixed FP64 encoder implementation, 16-byte code and four-mode weighting of Appendix B are unchanged; the rows are independent, their modes are not.

The single family contains 144 one-sided statements: 72 weighted gains G, 24 gains g_{20} for predictors at $r = .5$, and 48 gains g_{10}, g_{20} for predictors at $r = .9$. The $r = 1$ conflict lower bound is declared zero; its predictor risk upper bounds are still evaluated. The other mode means are descriptive. No statistic, predictor, smoothing value or sample size is added after this evaluation.

For any included scalar row statistic Y, let $\overline{Y}$ be its mean, s_Y^2 its unbiased sample variance, $T = 65536$, $F = 144$, $\alpha = .05$, and $W = 128\log 2 - \log(.01)$. Applying Theorem 4 of Maurer & Pontil (2009) to the reflected, range-normalized variable and taking a union bound gives

$$L(Y) = \overline{Y} - \sqrt{\frac{2s_Y^2\log(2F/\alpha)}{T}} - \frac{7W\log(2F/\alpha)}{3(T-1)}.$$

With probability at least .95, all 144 population means are at least their respective $L(Y)$. The variance for G uses the complete weighted row, including covariance between modes. Independence among functions or fitted models is unnecessary. The radius is $0.0162555323681\, s_Y + 0.0287718517065$ nats.

For each code and required string, maximize $L(g_{j0})$ over its already covered attached predictors and then clip the resulting information lower bound at zero. Denote it by M_j^L. This selection is covered by the same finite-family event, and the selected predictor is recorded for each code and string. Set $B_M = \frac{1}{2}M_2^L$ at $r = .5$ and $B_M = \frac{3}{20}M_1^L + \frac{13}{20}M_2^L$ at $r = .9$. Let q_D be the paired direct encoder's co-trained predictor, with statistic G_D, and let Z_M, Z_D be the matching and direct codes. Every predictor q of the matching code then satisfies, on that event,

$$L(q, Z_M) - L(q_D^\tau, Z_D) \geq B_M + L(G_D) - 64 \log 2 \max(.75, r).$$

A positive right-hand side establishes the claimed comparison. Both terms are covered by this new event; the original candidate-score event is not combined with it.

C.3 Results for all initializations and availability probabilities

Table 4 reports all six condition groups; the numerical record retains all 18 pair rows and all 72 predictor bounds. A nonpositive paired margin leaves the risk ordering unresolved. At $r = 1$, the declared code-regret lower bound is zero and leaves the optimality of each fitted code unresolved.

Table 4: The complete secondary comparison, in nats per request. Each paired margin is R^* plus the code-regret lower bound minus the smoothed direct-risk upper bound. Displayed ranges span the three fixed seeds and are not confidence intervals. All bounds share the stated 95% event and use the 1% mixture over the complete masked target. Displayed endpoints are rounded outward; inference uses full precision.

r	Start	Code-regret lower	Direct-risk upper	Paired margin
.5	Raw	[3.645, 3.652]	[11.205, 11.212]	[3.524, 3.537]
.5	Parity	[0.731, 0.743]	[22.302, 22.307]	[−10.481, −10.473]
.9	Raw	[6.090, 6.106]	[11.204, 11.207]	[−0.678, −0.663]
.9	Parity	[1.216, 1.224]	[4.535, 4.541]	[1.113, 1.124]
1	Raw	0	[11.191, 11.193]	[−11.193, −11.191]
1	Parity	0	[0.080, 0.081]	[−0.081, −0.080]

Independent reaggregation and bounded NumPy forward replay agreed with the saved evaluation; training was not replayed. The numerical record gives the exact replay coverage and discrepancies, and `numerical_support/replay_secondary_bounds.py` reconstructs all 144 covered statistics and selections, including negative pre-clipping values.

Bounds on the best achievable risk. Taking the smallest risk upper bound among a code's three covered smoothed predictors bounds its Bayes risk on the same event, without assuming an optimal predictor. Figure 2 shows the wide highlighted brackets; `numerical_support/bayes_risk_brackets.json` gives all 18 brackets and the selected upper-bound predictors. The full irrecoverable penalty remains unresolved.

C.4 Retained information and a change of availability

The same event also shows how the parity-start matching codes at $r = .9$ reallocate storage, and what that reallocation is worth when the other string is absent. Neither consequence adds a statistic, fitted model or confidence statement. Since $M_j \geq M_j^L$ and $M_j \leq aI(U_j; Z)$,

$$I(U_1; Z) + I(U_2; Z) \geq (M_1^L + M_2^L)/a.$$

For independent strings, $I(U_1; U_2 \mid Z) = D - I(U_1; Z) - I(U_2; Z)$. The code budget $D \leq 128 \log 2$ therefore gives

$$I(U_1; U_2 \mid Z) \leq 128 \log 2 - (M_1^L + M_2^L)/a.$$

Each parity-start matching code at $r = .9$ retains at least 20.12 bits in total about the strings separately. The conditional mutual information between the strings given each code is at most 107.88 bits, compared with 128 bits given the initial parity code. These bounds do not identify specific retained coordinates or determine which string contributes most of the actual information. The exact three-seed values are in `numerical_support/information_annotation_20260923/ALL_R090_WRITERS.csv`.

The same codes without the other string. Keep the six fixed pairs trained at $r = .9$ and set the availability to $r = 0$. The indicator E is independent of the source, encoder randomness, request and mask, so the joint law of (U_1, U_2, Z, J, S) given $E = 0$ is unchanged. The gains g_{j0} already bounded at $r = .9$ therefore describe the new workload, without reweighting any rows. Write $A = 64 \log 2$ and $R_0(Z)$ for a code's minimum risk at $r = 0$.

For the paired direct code Z_D, let $U_D = 64 \log 2 - L(G_D)$ be the simultaneous upper bound on its smoothed predictor's risk at $r = .9$, and put $\delta_D = U_D - A/10$. The regret inequality (7) at $r = .9$, applied to the direct code's own information, gives $.15M_1 + .65M_2 \le R_{.9}(Z_D) - A/10 \le \delta_D$. At $r = 0$, (3) makes the code's predictive information $.75M_1 + .25M_2$. Since $.75 = 5 \times .15$ and $.25 < 5 \times .65$, this is at most $5\delta_D$; five is the smallest uniform factor this coefficient comparison gives. Together with the optimum $A/4$ at $r = 0$,

$$R_0(Z_D) \ge \max\{A/4,\ A - 5\delta_D\}.$$

For the matching code Z_M, choose for each request j the attached predictor with the largest simultaneous lower bound $L(g_{j0})$, or the uniform distribution if every such bound is negative. The predictor q_M uses the chosen smoothed distribution for every request of string j, computed from the code, request and mask alone; these calls never receive the other string. The choice is fixed per request, not per target or example, so q_M is normalized. Its log loss at $r = 0$ is $A - .75\,\mathbb{E}g_{10} - .25\,\mathbb{E}g_{20}$ for the chosen predictors, and the uniform distribution has gain exactly zero. On the event, therefore,

$$L_0(q_M, Z_M) \le A - .75M_1^L - .25M_2^L.$$

The fallback is what makes clipping M_j^L at zero valid for this actual predictor. Every predictor q of the direct code then satisfies

$$L_0(q, Z_D) - L_0(q_M, Z_M) \ge \max\{A/4,\ A - 5\delta_D\} - A + .75M_1^L + .25M_2^L.$$

Both inequalities use only statements already in the 144-statement family, so they hold on the same 95% event as the original comparison at $r = .9$. The request-wise construction of q_M from predictors included in the event is post hoc.

Table 5: All six pairs trained at $r = .9$, on the original simultaneous event, in nats per request. Advantages are lower bounds. At $r = 0$, the direct-code column bounds that code's minimum risk from below, and the matching column bounds the risk of the constructed predictor q_M from above. Lower bounds are rounded down and upper bounds up.

		$r = .9$	$r = 0$		
Start	Seed	Direct advantage	Direct-code risk	Matching risk	Matching advantage
Parity	210101	1.123	43.862	39.293	4.570
Parity	210102	1.113	43.842	39.302	4.540
Parity	210103	1.121	43.858	39.273	4.586
Raw	210101	−0.678	11.090	32.911	−21.821
Raw	210102	−0.664	11.090	32.867	−21.777
Raw	210103	−0.674	11.090	32.863	−21.772

For each parity-start pair, request 1 selects the refitted product predictor and request 2 the co-trained predictor; the fallback is not used (Table 5). Each of these direct codes has $\delta_D \le .104$ nats, while the matching codes give $M_1^L \ge 6.63$ and $M_2^L \ge .34$ nats. The minimum risks of the same frozen codes are therefore ordered oppositely at $r = .9$ and $r = 0$. The raw-start direct codes are far from optimal at $r = .9$ ($\delta_D \approx 6.77$ nats), so only the universal floor $A/4$ applies and

both margins are negative; these lower bounds are inconclusive, not evidence against a reversal. The optimum at $r = 0$ stores U_1 with risk $A/4 \approx 11.09$ nats. The matching bounds near 39.3 nats imply no closeness to it, and no Hamming proximity of the direct codes to parity is claimed. `numerical_support/availability_reversal/` gives the six rows at full precision and a standard-library replay from the 144 supplied statistics.

C.5 Repeating the experiment with a different straight-through gradient

The encoder of Appendix B uses the identity straight-through gradient through its hard signs. Let h be its vector of preactivations for one input row. The alternative backward map differentiates $h/\max\{\mathrm{RMS}(h), 10^{-8}\}$, retaining the same hard forward code and all other settings. Above the clamp, this derivative removes the radial component and rescales tangential components. It remains a biased straight-through estimator; the comparison changes the complete derivative, not only its radial part. The run does not record whether every training row stayed above the clamp.

The panel crosses the two maps with all three availabilities, both starts, both objectives and the original three seeds: 72 encoder/co-trained-predictor fits, and fresh product and AR predictors for each of the 36 matching codes. All 144 fits complete the original 8,192 updates from the same initial tensors and training streams. The 36 identity-map encoders and co-trained predictors are identical, tensor for tensor, to the original exports, so their results below re-evaluate the same models rather than add training seeds; their refitted fresh predictors were not compared with the original ones. Only the RMS arm trains new encoders.

Evaluation uses a fresh source/mask stream and two new events, neither combined with the events above. The predictive event applies Appendix C.2's construction to all 72 codes and 144 attached predictors: 144 weighted gains, 48 gains g_{10} at $r = .9$ and 96 gains g_{20} at $r \in \{.5, .9\}$, so $F = 288$. The matching event applies the construction of Appendix A to all 72 co-trained critics with fresh candidate sets ($M = 72$, radius 0.0859336549 nats). Each event holds with probability at least .95; by a union bound both hold together with probability at least .90.

Table 6: Direct prediction versus every predictor of the matching code on the new predictive event, in nats per request. Ranges span the three seeds and are rounded outward; they are not confidence intervals. "Positive" counts seeds with a positive lower margin. Identity-gradient codes are the original models, evaluated on fresh rows.

		Identity gradient		RMS gradient	
r	Start	Paired margin	Positive	Paired margin	Positive
.5	Raw	$[3.530, 3.541]$	3/3	$[3.796, 3.804]$	3/3
.5	Parity	$[-10.487, -10.472]$	0/3	$[-10.455, -10.440]$	0/3
.9	Raw	$[-0.691, -0.669]$	0/3	$[-0.108, -0.092]$	0/3
.9	Parity	$[1.101, 1.103]$	3/3	$[1.080, 1.091]$	3/3
1	Raw	$[-11.204, -11.201]$	0/3	$[-11.203, -11.199]$	0/3
1	Parity	$[-0.097, -0.096]$	0/3	$[-0.097, -0.096]$	0/3

Under the RMS map, all three seeds of both originally successful conditions keep positive margins: at least 3.796 nats for the raw start at $r = .5$ and 1.080 nats for the parity start at $r = .9$ (Table 6). The other four groups remain inconclusive under both maps. The availability reversal above, declared before this panel was trained or evaluated, gives matching-code advantages at $r = 0$ of at least 4.534 nats for the RMS parity-start pairs and 4.484 nats for the identity pairs. All six raw-start reversal lower bounds are negative and leave the ordering unresolved.

On the matching event, the same three matching conditions as in the original primary event have positive code-regret lower bounds in every seed under each map: at least 2.322/2.524 nats for the raw start at $r = .5$, 1.021/1.512 at $r = .9$, and .614/.632 for the parity start at $r = .9$ (identity/RMS). All other matching codes and all 36 direct-trained codes have zero lower bounds.

The derivative does not rescue the difficult starts. Direct prediction from parity at $r = .5$ and from the raw start at $r = .9$ and $r = 1$ keeps smoothed empirical losses of about 22.26, 11.13 and 11.12 nats

under both maps, against optima 11.09, 4.44 and 0; the RMS changes are millinats. RMS lowers the empirical losses of fresh predictors for raw-start matching codes by about 1.8–3.0 nats. Nevertheless, none of the 72 prescribed comparisons places an RMS predictor's simultaneous risk upper bound below the lower bound for every predictor of the corresponding identity-gradient code. A larger regret lower bound under one map likewise does not show larger actual regret.

Independent checks reaggregated all saved predictive rows and candidate sets; reconstructed all 288 bounds, 36 pair margins, twelve reversals and 72 map comparisons; and replayed the first 16 rows of every code with a separately written forward pass (largest discrepancy 6.1×10^{-14}). Training was not replayed. FP32-to-FP64 export changed 24 of 603,979,776 tested code bits across 17 encoders; the fixed FP64 export is the evaluated code. The panel compares two specific surrogate gradients for one source with supplied parity features, trained for 8,192 updates. It does not isolate radial deletion or establish robustness to other optimizers or schedules. `numerical_support/rms_derivative/` supplies the evaluated statistics, all pair, reversal, matching-event and map-comparison rows, and a standard-library replay.

D GAUSSIAN MATCHING AND EXACT FINITE ALLOCATION

The source and rank budget are those of Section 4. This appendix specifies the code-candidate orientation and proves Theorem 3, including all optimal rowspaces. Gaussian-mixture analyses study Fisher-subspace recovery for population dot-product objectives under appropriate augmentation and dimension conditions (Bansal et al., 2025). Here the candidate set is finite, the critic is Bayes optimal, and several requests share a rank-constrained code. For conditional InfoNCE, draw V, sample $Y_1, \dots, Y_K$ iid from $P(Y \mid V)$, choose a uniform index I, and draw $X \sim P(X \mid V, Y_I)$. The critic receives X, V and the candidate codes $Z_k = QY_k$, $k = 1, \dots, K$, and predicts I. The candidate count includes the positive. Its optimal value is

$$J_K(Z) = \log K - H(I \mid X, V, Z_1, \dots, Z_K) \le \log K. \tag{8}$$

The entropy averages over sampled candidate sets. We call $\log K - J_K(Z)$ the *matching deficit*; it measures the remaining candidate ambiguity. Both the critic and predictor are optimized for every encoder, so the objectives can differ in which encoder they select. Negatives use the true conditional marginal: every candidate in a set is drawn conditional on the same realized request V.

For a Gaussian target–code pair with squared canonical correlations u_l, let $\mu_l = -\frac{1}{2}\log(1-u_l)$. The optimized matching value can be written as the following finite Gaussian channel:

$$\Phi_K(\mu) = I(I; O \mid B_1, \dots, B_K), \qquad O_l = \sqrt{e^{2\mu_l} - 1}\, B_{I,l} + N_l, \tag{9}$$

with iid standard-normal entries in the candidate vectors B_j and noise N. The critic observes the candidate vectors $B_1, \dots, B_K$; conditional on them, the unknown input is one of these K vectors. This finite experiment has a different information reward from predicting the full Gaussian target. We retain the scalar notation $j_K(a) = \Phi_K(-\frac{1}{2}\log(1-a))$.

D.1 FINITE-CANDIDATE ALLOCATION WITH UNEQUAL WITHIN-BLOCK STRENGTHS

D.1.1 THE FULL-VECTOR CANDIDATE PROFILE

For the channel in (9), write $\mathcal{B} = (B_1, \dots, B_K)$, $T = O$, $s_i = u_i/(1-u_i)$ and $J_K(u) = \Phi_K(\mu) = I(I; T \mid \mathcal{B})$, where $\mu_i = -\frac{1}{2}\log(1-u_i)$ and $0 \le u_i < 1$. The expectation includes all iid Gaussian candidate sets. We first establish the curvature signs used to rule out fractional allocations. These signs let us derive the optimal blockwise rowspaces; a separate diminishing-gains argument then determines how many directions to retain from each block.

Lemma 6 (Curvature of the averaged Gaussian candidate profile)**.** *For every finite $K \ge 2$ and finite number q of coordinates, the profile is symmetric, unchanged by zero padding, and satisfies*

$$\partial_i J_K > 0, \qquad \partial_{ii} J_K > 0, \qquad \partial_{i\ell} J_K \le 0 \quad (i \ne \ell). \tag{10}$$

At zero coordinates these are right derivatives. The derivatives are in squared correlations u, rather than signal-to-noise ratios s.

Proof. Fix coordinate i, abbreviate $s = s_i$, and put $X = B_{I,i}$ and $\mathcal{G} = \sigma(\mathcal{B}, T)$. Define

$$e = X - \mathbb{E}[X \mid \mathcal{G}], \quad V = \mathbb{E}[e^2 \mid \mathcal{G}], \quad M = \mathbb{E}V, \quad R_2 = \mathbb{E}V^2, \quad F_4 = \mathbb{E}e^4.$$

Conditional on $(\mathcal{B}, T_{-i})$, the input law does not depend on s_i and N_i remains independent. Conditional I-MMSE and its derivative (Guo et al., 2005; 2011) therefore give

$$J_{s_i} = \tfrac{1}{2}M, \qquad J_{s_i s_i} = -\tfrac{1}{2}R_2, \qquad J_{u_i u_i} = \frac{(1+s)^3}{2}\{2M - (1+s)R_2\}. \tag{11}$$

Here derivatives with an s subscript refer to $J_K(s/(1+s))$. Finite noise makes every posterior candidate probability positive, and the Gaussian candidate coordinates are not all equal almost surely; hence $M > 0$. This proves the first sign. To prove the second sign, it remains to establish the strict bound $R_2/M < 2/(1+s)$. That bound requires averaging the Gaussian candidates, as the argument below shows.

A moment comparison after marginalization. Suppose $\alpha_\ell > 0$ and g is positive, bounded, log-concave and centrally even on $\mathbb{R}^q$. For $\rho(d) = e^{-\frac{1}{2}\sum_\ell \alpha_\ell d_\ell^2} g(d)$, we claim

$$\int d_i^4 \rho(d)\, dd \le \frac{3}{\alpha_i} \int d_i^2 \rho(d)\, dd. \tag{12}$$

Indeed, define

$$h_i(x) = \int_{\mathbb{R}^{q-1}} e^{-\frac{1}{2}\sum_{\ell \ne i} \alpha_\ell d_\ell^2} g(d_1, \dots, x, \dots, d_q)\, dd_{-i}.$$

Prékopa's marginalization theorem (Prékopa, 1973, Theorem 6) makes h_i log-concave. Central evenness and the change $d_{-i} \mapsto -d_{-i}$ make it even, so it is nonincreasing in $|x|$. For $G \sim N(0, \alpha_i^{-1})$, weight its law by $G^2/\mathbb{E}G^2$. Under this probability law, G^2 and $h_i(G)$ are oppositely monotone in $|G|$, and thus have nonpositive covariance. Consequently

$$\frac{\mathbb{E}[G^4 h_i(G)]}{\mathbb{E}[G^2 h_i(G)]} \le \frac{\mathbb{E}G^4}{\mathbb{E}G^2} = \frac{3}{\alpha_i},$$

which proves (12). Separate evenness in each coordinate before marginalization is unnecessary.

The exact pair measure keeps the full denominator. Set $S = \operatorname{diag}(\sqrt{s_1}, \dots, \sqrt{s_q})$. For a fixed candidate set, let $L_a(t) = \phi_q(t - Sb_a)$ and $p_a(t) = L_a(t)/\sum_c L_c(t)$. Since the observation density is $K^{-1}\sum_c L_c(t)$, for $a \ne b$ and $m = 1, 2$,

$$\mathbb{E}[p_a(T)p_b(T)(B_{a,i} - B_{b,i})^{2m}] = \frac{1}{K}\mathbb{E}_{\mathcal{B}} \int (b_{a,i} - b_{b,i})^{2m} \frac{L_a(t)L_b(t)}{\sum_c L_c(t)}\, dt. \tag{13}$$

Change variables to $c = (b_a + b_b)/2$, $d = b_a - b_b$, and $w = t - Sc$. The Gaussian candidate distribution makes $c \sim N(0, I_q/2)$ and $d \sim N(0, 2I_q)$ independent. Holding c, w and all remaining candidates fixed, define

$$b(d) = \tfrac{1}{8}\|Sd\|^2, \qquad h(d) = \tfrac{1}{2}w^T Sd, \qquad C = \sum_{v \notin \{a,b\}} \frac{\phi_q(w + S(c - b_v))}{\phi_q(w)}.$$

The quantity $C \ge 0$ is independent of the entire vector d. Completing the Gaussian squares gives the exact identity

$$\frac{L_a L_b}{\sum_c L_c} = \phi_q(w) \frac{e^{-b(d)}}{e^{h(d)} + e^{-h(d)} + Ce^{b(d)}}. \tag{14}$$

Multiplication by the Gaussian density of d leaves, up to a positive factor independent of d,

$$\rho(d) = \exp\left[-\sum_\ell \frac{2 + s_\ell}{8} d_\ell^2\right] g(d), \qquad g(d) = \{e^{h(d)} + e^{-h(d)} + Ce^{b(d)}\}^{-1}.$$

This g is positive, bounded by $1/2$, and centrally even. It is log-concave because its negative logarithm is log-sum-exp of affine functions and a positive semidefinite quadratic; when $C = 0$, omit the last term. Applying (12) with $\alpha_i = (2 + s_i)/4$, then integrating the external variables, bounds each

pair's fourth moment by $12/(2+s_i)$ times its second moment. All candidates and all informative coordinates have remained in this calculation.

For *unordered* posterior pairs, the central-moment identities are

$$\sum_{a<b} p_a p_b (b_{a,i}-b_{b,i})^2 = V, \qquad \sum_{a<b} p_a p_b (b_{a,i}-b_{b,i})^4 = \mathbb{E}[e^4 \mid \mathcal{G}] + 3V^2.$$

Summing the pair bound and averaging therefore yields

$$F_4 + 3R_2 \le \frac{12M}{s+2}. \tag{15}$$

Conditioning first on the other observations would give unequal index weights. The bound above is obtained by averaging the full experiment; it is not a pointwise claim about arbitrary posterior weights.

Gaussian noise closes the curvature bound. Put $V_3 = \mathbb{E}[e^3 \mid \mathcal{G}]$. Posterior differentiation gives the identities (Dytso et al., 2023, Proposition 7) $\partial_{T_i}\mathbb{E}[X \mid \mathcal{G}] = \sqrt{s}V$ and $\partial_{T_i} V = \sqrt{s}V_3$. Integration by parts in the actual noise, with the candidate set and actual index held fixed, gives

$$\mathbb{E}[N_i e^3] = -3\sqrt{s}R_2, \qquad \mathbb{E}[N_i^2 e^2] = M + 2sR_2 - 2s\mathbb{E}[eV_3] = M + 2sR_2.$$

The last term vanishes by conditional centering. Cauchy–Schwarz for $N_i e$ and e^2, followed by (15), implies

$$9sR_2^2 \le (M+2sR_2)F_4 \le (M+2sR_2)\left(\frac{12M}{s+2} - 3R_2\right).$$

For $s > 0$, put $r = R_2/M$. Rearrangement gives

$$P_s(r) := 15s(s+2)r^2 + (6-21s)r - 12 \le 0, \qquad P_s\left(\frac{2}{1+s}\right) = \frac{6s(s+11)}{(1+s)^2} > 0.$$

This quadratic has exactly one positive root. Hence $R_2/M < 2/(1+s)$, and (11) proves $J_{u_i u_i} > 0$.

Zero coordinates and mixed derivatives. At $s_i = 0$, the posterior weights p based on the other coordinates are independent of the current candidate column $x \sim N(0, I_K)$. Set $B = \operatorname{diag}(p) - pp^T$ and $\tau = \operatorname{tr} B$. Then $V = x^T B x$, $\mathbb{E}[V \mid p] = \tau$, and $\mathbb{E}[V^2 \mid p] = \tau^2 + 2\operatorname{tr}(B^2)$. Gershgorin's bound gives $0 \preceq B \preceq I/2$, while positive posterior weights give $0 < \tau < 1$. Thus

$$J_{u_i u_i}(0) = \tfrac{1}{2}\mathbb{E}[2\tau - \tau^2 - 2\operatorname{tr}(B^2)] \ge \tfrac{1}{2}\mathbb{E}[\tau(1-\tau)] > 0.$$

The Gaussian-channel information Hessian (Payaró & Palomar, 2009, Theorem 5), conditional on the candidate set and then averaged, gives for $i \ne \ell$

$$J_{s_i s_\ell} = -\tfrac{1}{2}\mathbb{E}[\operatorname{Cov}(B_{I,i}, B_{I,\ell} \mid \mathcal{G})^2] \le 0.$$

Changing to u multiplies this by $(1+s_i)^2(1+s_\ell)^2$, preserving its sign. Permuting iid candidate columns proves symmetry; zero coordinates add independent noise. For finite K, q, posterior moments are bounded by polynomials in the Gaussian candidate maxima, which have all finite moments. These bounds justify differentiation and integration by parts on compact finite-SNR intervals, and the conditional MMSE formulas and continuity give the right derivatives at zero. □

D.1.2 Exact allocation and every maximizing rowspace

For Section 4's independent source blocks and requests, write $\pi_j = p_j$ and allow a signal matrix $0 < A_j < I$:

$$X^{(j)} = A_j^{1/2} Y^{(j)} + (I - A_j)^{1/2}\epsilon_j,$$

with independent standard Gaussian noise. Let $a_{j1} \ge \cdots \ge a_{jn_j} > 0$ be its eigenvalues and $f(a) = -\frac{1}{2}\log(1-a)$. The pre-request code and code-candidate experiment are those defined above.

Theorem 7 (Finite-candidate allocation)**.** *Fix finite $K \geq 2$, $0 \leq \lambda < \infty$, and an integer rank cap $0 \leq R \leq N := \sum_j n_j$. Among all deterministic linear codes of rank at most R, the maximum of Bayes matching information plus λ times retained predictive information is*

$$\max_{\substack{0 \leq k_j \leq n_j,\ k_j \in \mathbb{Z} \\ \sum_j k_j = R}} \sum_j \pi_j \left[J_K(a_{j1}, \ldots, a_{jk_j}) + \lambda \sum_{\ell \leq k_j} f(a_{j\ell}) \right]. \tag{16}$$

Every maximizing rowspace has rank R, separates the source blocks, and retains a top-k_j eigenspace in each block, for counts maximizing (16)*. Conversely, every such rowspace is optimal. There is the usual subspace freedom at a cutoff eigenvalue tie. The counts are obtained by selecting the largest weighted marginal gains, as proved below; cross-block ties can give multiple allocations. In the notation of Theorem 3, $B_j(k) = J_K(a_{j1}, \ldots, a_{jk})$ and $m_{jk} = f(a_{jk})$.*

Proof. **Canonical reduction and orientation.** Remove dependent output rows and whiten a rank-r code to QY with $QQ^T = I_r$. Its rowspace projector is $P = Q^T Q$. In request j, the squared canonical correlations of target and code are

$$u_j = \mathrm{eig}(A_j^{1/2} P_{jj} A_j^{1/2}).$$

Whitening the Gaussian candidate codes and applying a singular-value decomposition to the target–code cross covariance gives exactly the channel of Lemma 6; independent uninformative coordinates can be dropped. These invertible transformations preserve the candidate experiment. The request's objective is therefore $G(u_j) = J_K(u_j) + \lambda \sum_i f(u_{j,i})$.

The function G is symmetric and strictly increasing. On a two-coordinate line of fixed sum, $h(t) = G(\ldots, c+t, \ldots, c-t, \ldots)$ is even and

$$h''(t) = G_{ii} + G_{\ell\ell} - 2G_{i\ell} > 0.$$

Thus nontrivial pairwise averaging strictly lowers G: it is strictly Schur-convex in squared correlations. It is also increasing under weak majorization. Indeed, if $x \prec_w y$, choose z with $x \leq z \prec y$, and use $G(x) \leq G(z) \leq G(y)$.

For positive semidefinite A, C with decreasing spectra a, τ, the classical product inequality recalled by Tao et al. (2022, Section 1) gives

$$\mathrm{eig}(A^{1/2} C A^{1/2}) \prec_w a \circ \tau.$$

For completeness, submultiplicativity of the norm of the kth exterior power of $A^{1/2}C^{1/2}$ bounds the products of its top singular values by those of the two factors. Squaring gives eigenvalue log-majorization; the increasing convex exponential gives weak majorization. Singular C follows by continuity.

Pair transfers, attainment, and equality. Let τ_j be the decreasing spectrum of P_{jj}. The actual objective is at most

$$H(\tau) = \sum_j \pi_j G(a_j \circ \tau_j), \qquad 0 \leq \tau \leq 1, \quad \sum_{j,\ell} \tau_{j\ell} = r.$$

Dropping the within-block ordering constraints gives a hypersimplex relaxation. Its interior points need not be feasible projector spectra. Every nonvertex has at least two fractional coordinates, since r is integer. Transfer mass between two such coordinates while keeping their sum fixed. Within one block the second derivative along the transfer is

$$\pi_j \big\{ a_i^2 J_{ii} + a_\ell^2 J_{\ell\ell} - 2a_i a_\ell J_{i\ell} + \lambda [a_i^2 f''(a_i \tau_i) + a_\ell^2 f''(a_\ell \tau_\ell)] \big\} > 0.$$

For coordinates in different blocks it is the sum of their positive diagonal terms. An interior point of this nontrivial segment is therefore strictly worse than at least one endpoint. Every relaxed maximizer is a $0/1$ vertex. Every such vertex is realized by selecting the corresponding eigenvectors in the separate source blocks, and its relaxed value equals its actual value. The relaxed and actual maxima coincide.

This also controls *every* actual optimum. A fractional block spectrum would have an orientation upper bound strictly below the attained relaxed maximum. Thus every P_{jj} at an optimum is a projector. From $P^2 = P$,

$$P_{jj} - P_{jj}^2 = \sum_{v \neq j} P_{jv} P_{vj} = 0.$$

Every summand is positive semidefinite, so $P_{jv} = 0$ for $v \neq j$: the rowspace separates the blocks. For a block projector of rank k_j, the nonzero canonical eigenvalues are Ritz values of A_j, bounded termwise by its top k_j eigenvalues. Strict monotonicity forces equality. The equality case of the Ky Fan trace maximum gives the claimed top eigenspace, including cutoff-tie freedom.

This proves (16) at each fixed rank r. If $r < R$, a maximizing coordinate code leaves an informative direction unused. Adding it strictly improves matching and cannot reduce prediction. The rank-at-most-R optimum therefore uses rank R. □

D.1.3 Why greedy allocation is exact

For one block, let $O_\ell = (B_{1,\ell}, \ldots, B_{K,\ell}, X_\ell)$ be a complete coordinate observation of quality a_ℓ, ordered decreasingly, with $X_\ell = \sqrt{a_\ell} B_{I,\ell} + \sqrt{1 - a_\ell} N_\ell$. These observations are conditionally independent given I. Put $B(k) = I(I; O_1, \ldots, O_k)$ and $\delta_k = B(k) - B(k-1)$. Higher quality Blackwell-dominates lower quality (Blackwell, 1951): keep the candidate column and replace the query by

$$X_{\text{low}} = \sqrt{a_{\text{low}}/a_{\text{high}}} X_{\text{high}} + \sqrt{1 - a_{\text{low}}/a_{\text{high}}} N'.$$

This index-independent kernel has exactly the lower-quality law. For an independent fresh coordinate O'_k of quality a_k and $H = O_{1:k-1}$, conditional data processing and conditional independence give

$$\begin{aligned} \delta_{k+1} &= I(I; O_{k+1} \mid H, O_k) \\ &\leq I(I; O'_k \mid H, O_k) = \delta_k - I(O_k; O'_k \mid H) < \delta_k. \end{aligned}$$

To see strictness, take $g(O) = B_1 X$. Given I, $\mathbb{E}[g(O) \mid I] = \sqrt{a_k}\, \mathbf{1}_{\{I=1\}}$. If $p_1 = P(I = 1 \mid H)$, finite positive noise gives $0 < p_1 < 1$, so

$$\operatorname{Cov}(g(O_k), g(O'_k) \mid H) = a_k p_1 (1 - p_1) > 0.$$

The two observations are therefore not conditionally independent given H, establishing the strict information term.

The gains $\pi_j[\delta_{j,k} + \lambda f(a_{jk})]$ are positive and strictly decreasing within each block. Selecting the R largest gains automatically selects a prefix in every block and solves (16). Cross-block ties may be resolved in any way that selects R gains. This is the usual greedy argument for separable diminishing returns; it does not require joint convexity or interpolation between deterministic integer allocations.

D.1.4 Null signal directions and scope

For $0 \leq A_j < I$, let $N_+ = \sum_j \operatorname{rank} A_j$. In (16), use only positive eigenvalues and replace the total selected count by $\min(R, N_+)$. If $R \leq N_+$, every optimum has rank R, stores no null direction, and has the same block and top-eigenspace form. A relaxed optimizer cannot put positive trace on a zero-quality coordinate while a positive-quality coordinate remains below one: transferring trace to the latter strictly improves the value. The preceding transfer and equality proof then applies to the positive coordinates.

If $R > N_+$, every optimal rowspace contains the full informative space $\bigoplus_j \operatorname{range} A_j$ and may add any subspace of its orthogonal complement up to the cap. Its rank may be any integer from N_+ to R; extra null directions may mix across blocks. For necessity, use $A_j^{1/2} P_{jj} A_j^{1/2} \preceq A_j$. Strict profile monotonicity forces equality of all informative eigenvalues at the saturated optimum. Equality of traces gives $(I - P)A_j^{1/2} = 0$, with A_j embedded in its source block. Sufficiency follows immediately. In particular, a lower-dimensional query $X^{(j)} = B_j Y^{(j)} + \eta_j$, with independent noise covariance $\Sigma_j > 0$, has the operator $A_j = B_j^T (B_j B_j^T + \Sigma_j)^{-1} B_j$ and is covered by this clause.

The strict-noise assumption matters. With $A = \operatorname{diag}(1, .8, .2)$, rank cap two and $\lambda = 0$, retaining the first coordinate identifies the positive candidate almost surely. Both the rank-one first-coordinate code

and the code retaining the first and third coordinates attain $\log K$; full-rank and top-two conclusions would fail.

The curvature lemma is likewise averaged over iid Gaussian candidate sets. For the fixed binary candidate set $\{-2, +2\}$, (11) gives $J_{uu}(0) = 4 - 16/2 = -4$.

Nor is the vector profile jointly convex: at zero,

$$\nabla^2 J_K(0) = \frac{K-1}{2K} I_q - \frac{K-1}{2K^2} \mathbf{1}\mathbf{1}^T,$$

whose all-ones eigenvalue is negative when $q > K$. Only the displayed pair transfers are used. Arbitrary additional side information does not preserve the strict signs; revealing the index, for example, leaves no matching information to gain. The theorem concerns independent orthogonal Gaussian demands, a shared deterministic linear code, iid marginal candidates, and the population Bayes critic.

D.2 A FINITE PREDICTIVE WEIGHT

Consider two independent n-dimensional blocks of equal quality $a \in (0, 1)$, request probabilities $(p, 1-p)$ with $p > 1/2$, and linear codes of rank at most n. Set $m = -\frac{1}{2}\log(1-a)$ and let $B_K(k) = \Phi_K(m\mathbf{1}_k)$, $B_K(0) = 0$, and $\delta_k = B_K(k) - B_K(k-1)$ for this source. The predictive regret is $\Delta_n(Q) = pnm - I(X; QY \mid V)$. For $C_K(Q) = \log K - J_K(Q)$ define $F_\lambda(Q) = C_K(Q) + \lambda\Delta_n(Q)$ and its excess $\xi_\lambda(Q) = F_\lambda(Q) - \min F_\lambda$. All quantities use summed nats and the same stored code.

Corollary 8 (Finite threshold)**.** *The first-block code is optimal precisely when*

$$\lambda \geq \lambda_{\mathrm{crit}} := \left[\frac{(1-p)\delta_1 - p\delta_n}{(2p-1)m}\right]_+. \tag{17}$$

It is the unique optimal rowspace for $\lambda > \lambda_{\mathrm{crit}}$, and every linear code in this class then satisfies

$$\Delta_n(Q) \leq \frac{\xi_\lambda(Q)}{\lambda - \lambda_{\mathrm{crit}}}. \tag{18}$$

Proof. Theorem 7 reduces every optimum to k minority directions and gives positive, strictly decreasing δ_k. The objective to maximize is $pB_K(n-k) + (1-p)B_K(k) + \lambda m[p(n-k) + (1-p)k]$. Its increment at k is $(1-p)\delta_{k+1} - p\delta_{n-k} - \lambda(2p-1)m$, strictly decreasing in k. Thus the first increment decides whether $k = 0$ is optimal. If the raw numerator in (17) is nonnegative, counts zero and one tie at its untruncated threshold; if negative, zero is uniquely optimal already at $\lambda = 0$. The all-optima result rules out extra mixed rowspaces.

Let Q_0 store the first block. Its optimality at the threshold implies $C_K(Q) - C_K(Q_0) + \lambda_{\mathrm{crit}}\Delta_n(Q) \geq 0$ for every code, including rank-deficient and mixed ones. Above the threshold, $\xi_\lambda(Q) \geq (\lambda - \lambda_{\mathrm{crit}})\Delta_n(Q)$. At a nonnegative untruncated threshold, the one-transfer code has positive regret and zero excess, so this strict inequality on λ is needed. For $n = 1$ the numerator is negative. At $p = 1/2$ the denominator vanishes: prediction ties across full-rank separated allocations while matching selects balanced counts. Relabel the blocks if $p < 1/2$. □

The increments sum to at most $\log K$, hence $\delta_n \to 0$ and $\lambda_{\mathrm{crit}} \to (1-p)\delta_1/[(2p-1)m] > 0$. At $a = .5, p = .75, K = 2$, numerical thresholds for $n = 5, 6, 32, 64, 128$ are $0, .043989, .226737, .227257, .227258$. The saved evaluation uses adaptive FP64 quadrature on $[0, 45]$ of the positive integral in Appendix G.5; estimated errors are not directed enclosures. Placing coefficient β on the dimension-averaged predictive loss alone corresponds to $\lambda = \beta/n$; dividing both losses by n leaves the relative weight unchanged. A fitted training loss does not supply the population excess, evaluated with the Bayes critic and predictor, that the corollary requires.

E PREDICTION REGRET AT A POPULATION MATCHING TOLERANCE

Let $Y^{(1)}, Y^{(2)}$ be independent n-dimensional standard Gaussians. A rank-at-most-n code is written before a request with probabilities $(p, 1-p)$, $p > 1/2$. Its target is $X = \sqrt{a}Y^{(j)} + \sqrt{1-a}N$ in

request j, with $0 < a < 1$. Keep a, p and integer $K \geq 2$ fixed as n grows. Put

$$m = -\tfrac{1}{2}\log(1-a), \qquad \gamma = \tfrac{1}{2}\log\frac{2-a}{2(1-a)}, \qquad M_j = I(X; QY \mid V = j).$$

The optimal predictive code retains the first block. Define

$$\Delta_n = pnm - pM_1 - (1-p)M_2, \qquad R_n(\eta) = \inf_{J_K^* - J_K(Q) \leq \eta} \Delta_n(Q). \tag{19}$$

Here $J_K^* = \max_Q J_K(Q)$, with both optimizations taken over the same rank-constrained class. Thus $R_n(\eta)$ is the least predictive regret among codes whose population matching gain is within η of optimal.

Theorem 9 (Equal-quality precision)**.** *Every exact matching optimum satisfies*

$$M_j = nm/2 + O(1), \quad nm - M_1 - M_2 = O(1), \quad \Delta_n = (p - 1/2)nm + O(1). \tag{20}$$

If $T_n = \min\{\log(1/\eta_n), \gamma n/2\} \to \infty$*, then*

$$R_n(\eta_n)/T_n \longrightarrow (2p-1)m/\gamma. \tag{21}$$

At each fixed positive tolerance some feasible codes have bounded regret. These are population-value statements, not training rates.

E.1 Canonical coordinates and the feasible information region

Whiten a rank-$r \leq n$ encoder to $QQ^T = I_r$ and put $P = Q^TQ$. For the entropy lemma, allow request-specific qualities a_j and put $m_j = -\frac{1}{2}\log(1-a_j)$, $\gamma_j = \frac{1}{2}\log[(2-a_j)/(2(1-a_j))]$. Let $t_{j\ell}$ be the eigenvalues of P_{jj}. The squared canonical correlations in context j are the nonzero eigenvalues of a_jP_{jj}, with zeros added to length n. Define $f(u) = -\frac{1}{2}\log(1-u)$ and $g(u) = \frac{1}{2}\log\{(2-u)/(2(1-u))\}$. Then $M_j = \sum_l f(a_jt_{jl})$ and $G_j = \sum_l g(a_jt_{jl})$. The projector constraints give $0 \leq t_{jl} \leq 1$ and total trace $r \leq n$. Convexity and $f(0) = 0$ give $f(a_jt) \leq tm_j$. Also $g(u) = H(f(u))$, with $H(x) = \frac{1}{2}\log((e^{2x}+1)/2)$ convex and $H(0) = 0$. Its chord on $[0, m_j]$ gives $g(u) \leq \gamma_j f(u)/m_j$. Together, these inequalities give

$$G_j/\gamma_j \leq M_j/m_j \leq \operatorname{tr} P_{jj}, \qquad \sum_j M_j/m_j \leq n. \tag{22}$$

These inequalities also cover rank-deficient codes. For equal qualities, this also proves the predictive optimum used in (19).

Within each Gaussian context the standardized target–code canonical pairs are exchangeable. The optimal matching value obtained by sampling code candidates at a target therefore equals the value obtained by sampling target candidates at a code. This is a property of this source, not a general forward/reverse contrastive equivalence.

E.2 A uniform entropy comparison from classical testing

For every context j, we will prove

$$\frac{c_0e^{-G_j}}{\sqrt{1+G_j}} \leq D_{K,j} := \log K - J_{K,j} \leq (K-1)e^{-G_j}, \qquad c_0 = \frac{2\log 2}{\sqrt{6\pi}}. \tag{23}$$

For one context suppress j, write its squared correlations as u_l, and set $s_l = u_l/(1-u_l)$. The two-candidate Gaussian channel has independent noise-whitened coordinates with SNR s_l. Write $\overline{\Phi}$ for the standard normal upper-tail function. Given two candidate means m_0, m_1, Bayes binary error is $\overline{\Phi}(\|m_1 - m_0\|/2)$, including the coincident-mean case. Their random difference has independent coordinates $\mathcal{N}(0, 2s_l)$. Hence its error averaged over candidate sets is

$$\bar{P}_e = \mathbb{E}\overline{\Phi}(\sqrt{T}), \quad T = \sum_l \frac{s_l}{2}H_l^2, \quad H_l \overset{\text{iid}}{\sim} \mathcal{N}(0,1).$$

The Gaussian Laplace transform gives

$$B := \mathbb{E}e^{-T/2} = \prod_l (1 + s_l/2)^{-1/2} = e^{-G}, \quad G = \tfrac{1}{2}\sum_l \log(1 + s_l/2).$$

This is the Gaussian-input cutoff expression (Gallager, 1965). Under the tilt $d\nu = B^{-1}e^{-T/2}dP$, the normals remain independent with variances $(1 + s_l/2)^{-1}$, so $\mathbb{E}_\nu T = v := \sum_l u_l/(2 - u_l)$. The Gaussian tail bound $\overline{\Phi}(x) \geq e^{-x^2/2}/[\sqrt{2\pi}(1 + x)]$ for $x \geq 0$ gives

$$\bar{P}_e \geq \frac{B}{\sqrt{2\pi}}\mathbb{E}_\nu \frac{1}{1 + \sqrt{T}} \geq \frac{B}{\sqrt{2\pi}(1 + \sqrt{v})}.$$

The second step uses Jensen for $(1 + x)^{-1}$ and concavity of square root.

For binary posterior entropy in nats, the Hellman–Raviv bound (Hellman & Raviv, 1970) gives $D_2 \geq 2 \log 2\, \bar{P}_e$ after averaging over the Gaussian candidate sets. Thus

$$D_2 \geq \frac{2 \log 2}{\sqrt{2\pi}(1 + \sqrt{v})} e^{-G}. \tag{24}$$

For the upper, $\log(1 + t) \leq \sqrt{t}$ in the Bayes binary log-sum formula gives $D_2 \leq B$ via the swapped-hypothesis affinity. Dropping all but one negative gives $D_K \geq D_2$, while $\log(1 + \sum t_i) \leq \sum_i \log(1+t_i)$ gives $D_K \leq (K-1)D_2$. Since $g(u) = -\frac{1}{2}\log(1-u/(2-u)) \geq u/[2(2-u)]$, $v \leq 2G$ and $1 + \sqrt{2G} \leq \sqrt{3(1 + G)}$. This proves (23) with the displayed c_0. All steps permit arbitrary zero or heterogeneous correlations. In particular, no identical-spectrum or coordinate-encoder premise enters.

E.3 Proof of the precision theorem

Exact extrema exist: the feasible rowspaces form a finite union of compact Grassmannians, and positive noise makes predictive information and finite-index conditional entropy continuous in the rowspace. The exact regret decomposition is

$$\Delta_n = (2p - 1)M_2 + p(nm - M_1 - M_2) \geq (2p - 1)M_2.$$

For any total candidate deficit at most δ, the second context and (23) give $\delta \geq (1 - p)c_0 e^{-G_2}/\sqrt{1 + G_2}$. With $S = [\log((1 - p)c_0/\delta)]_+$, split into $G_2 \geq S$ and $G_2 < S$ to obtain

$$G_2 \geq [S - \tfrac{1}{2}\log(1 + S)]_+, \qquad \Delta_n \geq \frac{(2p - 1)m}{\gamma}[S - \tfrac{1}{2}\log(1 + S)]_+. \tag{25}$$

The logarithmic correction in this bound depends on the testing exponent, not the ambient dimension.

For a coordinate split with k directions, the swapped-hypothesis affinity is $e^{-\gamma k}$. Under its normalized geometric-mean tilt, the log ratio L has the law $d\sum_{l=1}^{k} A_l B_l$, with independent standard normals and $d = 2\sqrt{a/(2 - a)}$. Conditional on the A_l, this is normal with variance $W = d^2\chi^2_k$. The binary entropy is

$$D_2 = e^{-\gamma k}\mathbb{E}h(L), \quad h(l) = e^{l/2}\log(1 + e^{-l}), \quad \int_{\mathbb{R}} h(l)\, dl = 2\pi.$$

The integral follows by substituting $t = e^{-l}$ and integrating by parts. The conditional density is at most $(2\pi W)^{-1/2}$. For $k > 2$, Cauchy–Schwarz and $\mathbb{E}(\chi^2_k)^{-1} = 1/(k - 2)$ therefore give

$$D_2 \leq e^{-\gamma k}\frac{\sqrt{2\pi}}{d\sqrt{k - 2}}.$$

A coordinate code splitting its directions as evenly as possible between the blocks has weighted deficit $e'_n = O(n^{-1/2}e^{-\gamma n/2})$. The lower bound in (24) has a prefactor bounded below by a source constant times $n^{-1/2}$. At an exact optimum, its comparison with e'_n forces $G_j \geq \gamma n/2 - O(1)$. The bounds in (22) and $M_1 + M_2 \leq nm$ give $M_j = nm/2 + O(1)$ and nonnegative shortfall $O(1)$, proving (20).

For an η-near optimum use $\delta = \eta + e'_n$ in (25). If $\log(1/\eta) \le \gamma n/2$, then $-\log\delta = T_n + O(1)$. Otherwise it lies between $T_n - O(1)$ and $T_n + \frac{1}{2}\log n + O(1)$, with n proportional to T_n. Hence, uniformly in all regimes where $T_n \to \infty$,

$$R_n(\eta_n) \ge \frac{(2p-1)m}{\gamma} T_n - O(\log(1+T_n)).$$

For the upper, set $k = \lceil \log((K-1)/\eta_n)/\gamma \rceil$ when $k \le n/2$. A coordinate code with k directions from the second block has deficit at most η_n and exact regret $(2p-1)mk$. If $k > n/2$, then $\log(1/\eta_n) \ge \gamma n/2 - O(1)$ and an exact optimum supplies the upper with $O(1)$ remainder. This proves (21). A fixed positive tolerance requires only a fixed k, giving the final statement of the theorem.

F Common Gaussian optima and their limits

Proposition 10 (Conditional CCA)**.** *For jointly Gaussian (X, Y, V) with positive-definite covariance, the rank-r linear codes maximizing correctly conditioned finite-candidate matching are exactly those minimizing $h(X \mid V, QY)$, for every $K \ge 2$. The encoder observes Y only; the predictor and matching critic observe V after the code is made.*

Proof. For the jointly Gaussian triple in the proposition, let $S_X = \Sigma_{X|V} \succ 0$, $S_Y = \Sigma_{Y|V} \succ 0$ and $C = \Sigma_{XY|V}$. Put

$$F = S_X^{-1/2} C S_Y^{-1/2}, \qquad T = F^\top F.$$

Let $1 > \lambda_1 \ge \cdots \ge \lambda_n \ge 0$ be all eigenvalues of T, with zero padding when $n = \dim Y > \dim X$. For a full-row-rank Q, the rows of $U = (QS_YQ^\top)^{-1/2} QS_Y^{1/2}$ are orthonormal. Its retained canonical squared correlations are the eigenvalues η_j of $UTU^\top$, and Courant–Fischer gives $\eta_j \le \lambda_j$ for $j \le r$. Leading eigenvectors attain all these bounds with a single admissible Q.

Gaussian residuals $X - \mathbb{E}[X \mid V]$ and $Y - \mathbb{E}[Y \mid V]$ are jointly independent of V. Although the encoder cannot form the second residual itself, both the predictor and matching critic can form $QY - Q\mathbb{E}[Y \mid V]$. Under true conditional negatives, its Bayes candidate value is $\Phi_K(-\frac{1}{2}\log(1-\eta_1), \ldots, -\frac{1}{2}\log(1-\eta_r))$. I-MMSE makes this strictly increasing in every finite coordinate: an iid Gaussian candidate set has distinct entries almost surely and all finite-noise posterior weights are positive. Both candidate value and total conditional information therefore attain their optima exactly when $\eta_j = \lambda_j$ for every j. This proves equality of the entire optimal sets, including degeneracies. The optimal codes are exactly those whose conditionally whitened row space $\mathcal{S} = \mathrm{row}(QS_Y^{1/2})$ satisfies

$$E_{>\lambda_r}(T) \subseteq \mathcal{S} \subseteq E_{\ge\lambda_r}(T), \qquad \dim\mathcal{S} = r.$$

An eigengap at r gives uniqueness of this subspace.

□

F.1 A nearly optimal candidate code with common prediction demands

Let independent standard Gaussian vectors Y_1, Y_2, E have dimension 128, and let an independent request $V \in \{1, 2\}$ ask for the same target $X = (Y_1 + E)/\sqrt{2}$ in either case. The code $Z = Y_1$ is simultaneously optimal for prediction and every finite candidate count. Instead retain 100 coordinates of Y_1 and 28 coordinates of the irrelevant Y_2. This is an admissible orthonormal rank-128 projection. Its exact prediction regret is $28(\log 2)/2 = 14\log 2$ nats.

For a shared candidate index with $K = 1024$, maximum-likelihood decoding uses the 100 informative coordinates. The Gaussian pairwise tail bound, averaged over independent candidate means, is $(3/2)^{-50}$. A union bound gives average index error at most $e = 1023(3/2)^{-50}$. Writing h for binary entropy in nats, conditional Fano averaged over the candidate set gives

$$\log K - J_K(Z) \le h(e) + e\log 1023 \le e[1 + 50\log(3/2)] < 3.5 \times 10^{-5}.$$

For an exact conservative check, $e < 1.605 \times 10^{-6}$ is an integer-power comparison, and $\log(3/2) < .406$ follows from the first five positive terms of the exponential series. Their product bound is

.0000341865. Since the optimum is at most $\log K$, this proves the asserted near-optimality without computing that optimum.

The example separates exact optimizer alignment from quantitatively useful approximate calibration. It does not explain which update caused an actual learned trajectory to leave a good code, nor does it preclude a calibration modulus at fixed dimension. No competing demand or weak fitted predictor is needed for this near-optimum construction.

F.2 A POSITIVE REGIME SHARED BY PROPER TESTING LOSSES

For a positive-definite Gaussian target–observation pair, whiten both marginals. The retained squared canonical correlations of a rank-R linear code obey $\eta_i \le \lambda_i$, where the λ_i are the leading R full-view squared correlations. A top-CCA code attains them simultaneously. Under the positive pair distribution its canonical coordinates are correlated Gaussian pairs; under the product distribution they are independent standard Gaussians.

Starting from the top-CCA pair experiment, replace each code coordinate by

$$Z_i' = \sqrt{\eta_i/\lambda_i}\, Z_i + \sqrt{1-\eta_i/\lambda_i}\, N_i$$

with fresh independent standard normals, omitting zero λ_i. Rotations into the desired canonical bases complete one hypothesis-independent channel that maps both distributions to the competing pair experiment. This Blackwell dominance (Blackwell, 1951) gives a top-CCA optimum for every proper binary testing loss and every f-divergence. For strictly proper binary losses with finite risks, strict attenuation of a positive canonical correlation loses value; ties and infinite divergences require their own qualifications. Conditional averaging preserves the conclusion when one feasible raw-observation encoder attains the top conditional spectrum almost surely.

The comparison channel can transform target coordinates. This is dominance of the joint-versus-product *pair experiment*, not a code-only transformation preserving every particular raw-target task. The predictive agreement concerns full-target log loss in the stated Gaussian regime.

G GAUSSIAN NUMERICAL EVALUATION AND OPTIMIZATION COMPARISONS

G.1 BINARY PROFILE USED IN NUMERICAL EVALUATION

For a known Gaussian candidate set with two entries, subtract its midpoint and put $G = (B_1 - B_2)/\sqrt{2}$. The hidden-index problem becomes binary phase-shift keying (BPSK) at SNR $t = \frac{1}{2}\sum_\ell s_\ell G_\ell^2$, where $s_\ell = e^{2\mu_\ell} - 1$ is the coordinate SNR of Appendix E. Write $b(t)$ for binary-input mutual information. I-MMSE gives $b'(t) = \frac{1}{2}\,\mathrm{mmse}_{\mathrm{BPSK}}(t)$. Its posterior mean and a change of Gaussian variable yield

$$\mathrm{mmse}_{\mathrm{BPSK}}(t) = \mathbb{E}\,\mathrm{sech}^2(t + \sqrt{t}N) = e^{-t/2}\mathbb{E}\,\mathrm{sech}(\sqrt{t}N). \tag{26}$$

Indeed, the transformed integrand contains $e^x\,\mathrm{sech}^2 x$ against an even Gaussian weight; its even part is $\mathrm{sech}\, x$. Using the Fourier transform of sech,

$$\mathrm{mmse}_{\mathrm{BPSK}}(t) = \mathbb{E}_\nu e^{-Ut}, \qquad U = (1+W^2)/2, \qquad p_W(w) = \tfrac{1}{2}\,\mathrm{sech}(\pi w/2). \tag{27}$$

Here $U \ge 1/2$, $\mathbb{E}U = 1$, and U is nondegenerate. The characteristic function $\mathbb{E}e^{itW} = \mathrm{sech}\, t$ verifies the second moment. Integrating b' and averaging the independent Gaussian squares gives

$$\Phi_2(\mu) = \tfrac{1}{2}\mathbb{E}_\nu \frac{1-P(U)}{U}, \qquad P(u) = \prod_\ell (1+us_\ell)^{-1/2}. \tag{28}$$

Taking the BPSK signal to infinity gives $\frac{1}{2}\mathbb{E}_\nu(1/U) = \log 2$. Subtracting (28) from this identity yields the positive deficit integral used below. These are evaluations of the same Gaussian candidate experiment, with no fitted approximation to its critic.

G.2 A FINITE IMPLICATION OVER EVERY SUBSPACE

Take $n = 64, K = 2, (a_1, a_2) = (4/5, 3/10)$ and $p = 3/4$. Let $x_j = M_j/m_j$, $v_1 = pm_1$, $v_2 = (1-p)m_2$ and $D = \log 2 - J_2$. The secants give $x_1 + x_2 \le 64$ and $G_2 \le \gamma_2 x_2$. Block 2 has 64 squared canonical correlations, each at most $3/10$, so $\sum_\ell u_\ell/(2-u_\ell) \le 64(3/10)/(2-3/10) = 192/17$ uniformly over mixed subspaces. Equation (24) gives

$$D \ge \frac{1}{4}\frac{2\log 2}{\sqrt{2\pi}(1+\sqrt{192/17})}e^{-\gamma_2 x_2}.$$

For $x_2 \le 32$, use $\log 2 > .69$, $\sqrt{2\pi} < 2.51$, $\sqrt{192/17} < 3.37$ and $e^{-32\gamma_2} = (14/17)^{16}$. The resulting rational lower exceeds $.00140776 > .0014$. Thus $D \le .0014$ implies $x_2 > 32$ and

$$\Delta = v_1(64 - x_1 - x_2) + (v_1 - v_2)x_2 > 32(v_1 - v_2) = 12\log 5 - 4\log(10/7) > 17.88.$$

The last logarithmic inequality follows, for example, from the positive atanh series with a geometric remainder bound. Also $\operatorname{tr} P_{22} \ge x_2 > 32$. Information-equivalent rank and physical rank are otherwise distinct for mixed subspaces.

A 14/50 coordinate code is a feasible comparison. For $k > 1$ directions, the affinity-tilted log ratio has maximum density $\Gamma((k-1)/2)/(2d\sqrt{\pi}\Gamma(k/2))$. Integrating the entropy kernel from Appendix E gives

$$D_2(k,a) \le e^{-k\gamma(a)} \min\left\{1, \frac{\sqrt{\pi}\Gamma((k-1)/2)}{2\sqrt{a/(2-a)}\Gamma(k/2)}\right\}.$$

At $(14, .8)$ and $(50, .3)$ the weighted bound is approximately .000983970. For an exact conservative comparison, even-k gamma ratios reduce to central binomial coefficients; use $\pi < 22/7, 2\sqrt{.8/1.2} > 1.63$ and $2\sqrt{.3/1.7} > .84$. The rational upper is below $.000984807 < .00099$. Consequently a .0004 gap to the optimal matching value suffices for $D < .00139$, and hence the prediction implication. This argument uses the 14/50 code only as a feasible comparator; its exact optimality is established separately in Appendix G.5.

G.3 EARLIER COMPARISON WITH LEARNED AND BAYES CRITICS

An earlier six-fit study compared Bayes matching, learned Gaussian-quadratic matching and direct NLL at $p = .1$ and $.75$, using rank 64, two candidates and 8,192 Adam updates (Kingma & Ba, 2015). Its $p = .75$ Bayes endpoint missed $.0014$ even before uncertainty; the four-fit comparison below retains that threshold. Learned-critic error was measured separately from code regret. `numerical_support/appendix_records/gaussian_six_endpoints.csv` and the "Gaussian study" sections of the accompanying methods record retain all six endpoints, their exact training settings and critic-error measurements.

G.4 EXACT FINITE-CANDIDATE MEASUREMENT AND UNCERTAINTY

For candidate laws P_0, P_i, let $L_i = \log(P_0/P_i)$ and common pair affinity $B = \int \sqrt{P_0 P_i} = e^{-G}$. With $q = K - 1$, define $R_i = \sqrt{P_0 P_i}/B$ and $\bar{R} = q^{-1}\sum_i R_i$. Change of measure gives

$$D_K = qB\,\mathbb{E}_{\bar{R}}\frac{\log(1+\sum_i e^{-L_i})}{\sum_i e^{-L_i/2}}.$$

The integrand lies in $[0,1]$ because $\log(1+\sum t_i^2) \le \sqrt{\sum t_i^2} \le \sum t_i$. Candidate symmetry permits sampling R_1. For canonical target–code correlation matrix H, an exact Gaussian affinity sampler uses independent $z_+ \sim \mathcal{N}(0, I)$ and $z_- \sim \mathcal{N}(0, 2(I - H^T H)(2I - H^T H)^{-1})$, with $X \mid z_+ \sim \mathcal{N}(Hz_+/\sqrt{2}, I - HH^T)$. This follows by completing squares in $\sqrt{P_0 P_1}$ and remains valid at zero singular values.

For the earlier six endpoints, 65,536 independent paired-context proposals per endpoint give the supplied simultaneous upper bounds by Hoeffding's inequality with failure probability $.05/6$. Each weighted estimate lies in $[0, (K-1)(pB_1 + (1-p)B_2)]$. These bounds condition on fixed models and the stated mathematical sampling law. Numerical reconstruction uses FP64, without directed rounding. The separate four-endpoint event is specified below.

G.5 Population and sampled-candidate optimization

The four-fit comparison asks whether predictive regret persists near the optimal matching allocation. It compares population and sampled-candidate gradients at the same source, initialization and schedule, retaining the .0014 threshold.

What is integrated and what is learned. Use $n = 64$, $K = 2$, qualities $(.8, .3)$ and source seed 871901. All four fits use the earlier $p = .75$ initial orthonormal basis, converted to FP64, and optimize a full 128-by-64 raw matrix through signed QR. They share the same basis and raw scale at both $p = .75$ and $p = .1$. Each completes 8192 Adam updates at initial rate .002, cosine decay to 5% of that rate, betas $(.9, .999)$, epsilon 10^{-8}, and gradient clip 10. There is no coordinate restriction, spectral rounding, restart, checkpoint selection or outcome-dependent extension.

Both estimators know the source and use its Bayes critic. The sampled-candidate estimator draws one iid set of two candidates and 256 queries per context per update. The deterministic estimator integrates over the known Gaussian candidate law. For squared canonical correlations u_ℓ, put $s_\ell = u_\ell/(1 - u_\ell)$ and $U(w) = (1 + w^2)/2$. The binary deficit has the positive integral

$$D_2(u) = \frac{1}{2}\int_0^\infty \frac{\operatorname{sech}(\pi w/2)}{U(w)} \prod_\ell (1 + U(w)s_\ell)^{-1/2}\, dw.$$

The implemented objective and gradient use 128-node Gauss–Legendre quadrature on $[0, 24]$. The omitted value tail is bounded by $(4/\pi)e^{-12\pi} < 5.5 \times 10^{-17}$. Before fitting, we checked quadrature accuracy and gradients. The "Gaussian study: numerical instrument checks" section of the accompanying methods record gives these checks.

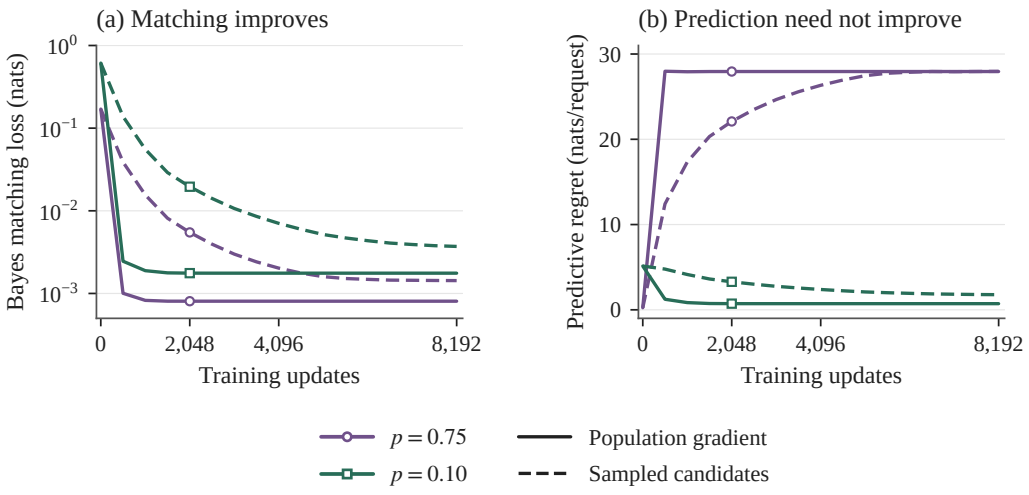

Figure 4: **Improving matching can either worsen or improve prediction.** Four Gaussian-encoder fits share one source rotation and initialization, using a known-source Bayes critic. Color and marker distinguish the request probability; solid and dashed curves use population and sampled-candidate gradients. At $p = .75$, predictive regret rises overall toward 28 nats while matching loss falls; at $p = .10$, both improve from the common start. Lines connect seventeen saved states per fit, at intervals of 512 updates; the markers identify the states at update 2,048. These are individual trajectories without smoothing or training confidence bands.

Exact reference and measured attainment. Theorem 7 reduces the global optimum to an integer allocation. The optimal strong/weak splits are 14/50 at $p = .75$ and 9/55 at $p = .1$, with numerical CE .000805979409 and .001759682608. The population-gradient endpoints attain those values to numerical precision. Table 7 reports all four endpoints; no finite-$K > 2$ fitted result is inferred.

Table 7: **Accurate population optimization can leave a large prediction cost.** One source rotation and initial orthonormal basis; all four fits complete 8192 updates. CE is numerical Bayes matching entropy in nats per candidate set. Parentheses give simultaneous 95% upper bounds from 65,536 independent tilted candidate proposals per endpoint, conditional on these four codes. Prediction regret is calculated from the saved subspace in nats per request.

p	Gradient	Bayes CE (95% upper)	Prediction regret
.75	Sampled candidates	.00143047 (.00145235)	27.96605
	Population	.00080598 (.00082141)	27.94774
.10	Sampled candidates	.00370628 (.00376260)	1.77312
	Population	.00175968 (.00179311)	.72029

The four frozen endpoints use the bounded affinity-tilted estimator above with 65,536 fresh independent paired-context proposals each, disjoint from training and previous readouts. Failure probability $.05/4$ per endpoint gives the simultaneous upper bounds in Table 7. Only the $p = .75$ population arm passes .0014. For this comparison, write $x_j = M_j/m_j$ and $v_j = p_j m_j$. The decomposition $\Delta = v_1(64 - x_1 - x_2) + (v_1 - v_2)x_2$ separates the information shortfall of mixed directions (the first term) from allocation to block 2 (the second). The population endpoint has $x_2 = 50$ and negligible first term. The paired sampled fit has 2.99412256 nats from the first term and 24.97192735 from the second. At $p = .1$, better optimization improves both objectives.

Separate NumPy QR, covariance-based information calculations, adaptive positive-entropy integration, and reconstruction of all saved readout blocks agree with the reported values to floating-point precision. This is accurate FP64 evaluation, not directed interval arithmetic.

The sampled-candidate fit is one stochastic realization. Relative to the historical study, FP64 also changes realized Gaussian draws; under Adam, changing the gradient estimator changes effective steps and epsilon behavior. The result isolates an attainable objective preference in this controlled source, without proving a sampling-noise-only explanation, general convergence, or an observation-only learning method.

G.6 Predictive supervision from observed pairs

In image-caption retrieval, Bleeker et al. (2023) use reconstruction of a pretrained sentence representation as a constraint alongside contrastive training. The comparison below keeps the Gaussian source and target fixed while varying the relative scale of predictive supervision.

This separate study uses two 64-dimensional Gaussian blocks of equal quality $.5$, request probabilities $(.75, .25)$, a rank-64 code and two candidates. Each of two paired seeds supplies 131,072 observed positive training pairs, split 98,304/32,768 by request. Their moments initialize the full-matrix encoder and quadratic matching critic. All ten arms complete 8,192 Adam updates without checkpoint selection. The initial prediction regrets are $.063259/.062623$ nats. Matching uses additional online candidate sets; prediction-only does not, so total training exposure is unequal.

The auxiliary loss is the negative log-likelihood of the empirical maximum-likelihood linear Gaussian predictor at the current code, with full residual covariance. Its gradient uses observed moments and request probabilities, without the source rotation, quality or population covariance. The Gaussian family remains stipulated and analytically fitted. A separate arm uses the population predictive gradient. Observed-row likelihood, gradient and training-interface checks agree with this implementation; this is an empirical Gaussian profile.

Empirical NLL/64 reduces code regret by 16.29%/17.90%, retaining 93.30%/91.24% of the paired population-gradient improvements. Summed empirical NLL reduces regret by 95.23%/95.26%, at a substantial matching cost. Prediction-only remains best for prediction, including with a separately fitted predictor (excess log losses $.06982/.07105$ nats per request). All eight endpoints trained with a matching term remain above the values of feasible comparators for the ideal population objective. This comparison leaves optimality for their empirical objectives unresolved and does not establish the $.0014$ matching-loss threshold for the unequal-quality source.

Table 8: All ten final codes; each slash-separated pair gives the fixed fitted seeds 890901/890902 in that order. Rows marked + add the named predictive loss to matching. Regret is optimal predictive excess in nats per request. Matching CE is in nats per two-candidate set; its columns use different evaluation samples. Bayes CE uses 65,536 tilted proposal units per code; co-trained and refitted critics use the same 1,024 previously evaluated candidate sets. Refitting uses the original training moments at each unchanged code. The later critic comparisons on the reused candidate sets are descriptive.

Objective	Regret	Bayes CE	Co-trained CE	Refitted CE
Matching	5.479/5.527	.00133/.00129	.0171/.0245	.00117/.00142
+ empirical NLL/64	4.586/4.538	.00227/.00233	.0200/.0236	.00206/.00257
+ empirical NLL	.2611/.2621	.1489/.1488	.7006/.7028	.1540/.1543
Empirical NLL only	.0211/.0223	.1712/.1711	—	.1779/.1792
+ population NLL/64	4.522/4.443	.00241/.00254	.0249/.0223	.00231/.00254

The later matching-critic comparison separates the code tradeoff from a co-trained critic's error. Each evaluation set contains two shared candidates and 64 queries per request; losses average within requests before applying weights $(.75, .25)$. Candidate sets are the independent sampling units. The summed-loss critics already classify 91.40%/91.26% correctly despite CE near $\log 2$. Reapplying the normalized Gaussian-likelihood matching critic at the final code lowers CE to $.15401/.15426$, near the Bayes values $.14778/.14774$ on the same draws. Refitted critics improve other arms too; the table compares every arm using the same procedure. The code regrets are unchanged. Thus the repair improves probability quality without eliminating the allocation tradeoff.

The original tilted evaluations have a simultaneous two-sided 95% event over ten fixed codes under exact arithmetic; evaluation uses FP64 without directed rounding. The later comparison reuses all original evaluation candidate sets and introduces no new training examples or encoder updates. Its uncertainty is descriptive, not a fresh confirmatory event. The supplementary directory `supplement/observed_pair_prediction` provides the full settings, all saved predictor and critic comparisons and their sampling qualifications.

H Uncapped objectives with later complementary information

Use the source, context V and randomized 2^n-valued encoder of Appendix A, now with arbitrary request probabilities $\pi_j > 0$, $\pi_1 + \pi_2 = 1$ (written p_j in Section 5), and request-specific retentions $a_j \in (0, 1]$. Conditional on $J = j$, mask coordinates are independent Bernoulli(a_j), independent of the source, encoder randomness and $E \sim \mathrm{Bernoulli}(r)$, $r \in [0, 1]$. The encoder still precedes J, E, S; any supplied other string is outside the budget $h = n \log 2$ nats. The target is $X = U_{J,S}$.

For a fixed context v, define the conditional joint-to-product ratio

$$L_v(z, x) = \frac{P_{ZX|v}(z, x)}{P_{Z|v}(z) P_{X|v}(x)}. \tag{29}$$

Zero-probability cells are omitted. In particular, when the other string is supplied, the denominator uses the code marginal conditional on that string. For divergence order $\alpha = 1 + t$, with $t > 0$, the pooled power moment is

$$F_t(Z) = \mathbb{E}_V \mathbb{E}_{P_{Z|V} P_{X|V}} L_V^{1+t} = \mathbb{E}_{V,Z,X} L_V^t. \tag{30}$$

The pooled power divergence $(F_t - 1)/t$ and pooled Rényi divergence $\log F_t / t$ (van Erven & Harremoës, 2014) have the same exact optimizers, but their optimization gaps have different units. Both pool the power moment before applying the final transformation. Averaging logarithms separately across contexts defines a different objective.

The ideal Pearson and Tsallis objectives in f-MICL use this power family (Lu et al., 2023). Pearson also appears in Relative Predictive Coding at $\alpha_{\mathrm{RPC}} = \beta_{\mathrm{RPC}} = 0$, $\gamma_{\mathrm{RPC}} = 1$ (Tsai et al., 2021, Section 2.2); the unskewed Rényi variational objective of Lee & Shin (2022, Eq. (4)) gives the same

exact ordering with an unrestricted critic. Their proposed methods use relative regularization or a skewed reference distribution to control estimation. Here the criterion uses the full conditional-product reference in (29), and the critic is unrestricted.

We compare these criteria with Bayes log loss $R(Z) = H(X \mid Z, V)$. Write $Q(Z) = I(X; Z \mid V)$ for the predictive information and $\Delta(Z) = R(Z) - \inf_{Z'} R(Z')$ for regret within the same encoder class. Optimization tolerances δ below are nonnegative.

H.1 Prediction

Put $D = I(U_1, U_2; Z)$ and $I_j = I(U_j; Z)$. We separate information about each string, their dependence conditional on the code, and unused code capacity through the normalized quantities

$$x_1 = I_1/h, \qquad x_2 = I_2/h, \qquad x_P = (D - I_1 - I_2)/h, \qquad x_0 = 1 - D/h. \tag{31}$$

They are nonnegative and sum to one: independence gives $D - I_1 - I_2 = I(U_1; U_2 \mid Z) \geq 0$, and the budget gives $D \leq H(Z) \leq h$. The coordinate x_P does not assert that the encoder literally computes coordinatewise parity.

Lemma 11 (Erasure and entropy bounds)**.** *Let U be a uniform n-bit string, W any jointly distributed variable, and S an independent mask retaining each coordinate with probability a. Then*

$$ah - H(U \mid W) \leq I(U_S; W \mid S) \leq aI(U; W).$$

For a uniformly chosen subset of fixed size s, the upper bound is $(s/n)I(U; W)$. Both statements hold conditionally whenever the string remains uniform and the mask remains independent under that conditioning, and may then be averaged over the conditioning variable.

Proof. The upper bound is the coordinate-chain argument in the proof of Proposition 4; for a uniform size-s subset, replace the inclusion probability a by s/n. For the lower bound, $H(U_A \mid W) \leq H(U \mid W)$ for every fixed A. Average over the mask and use $H(U_S \mid S) = ah$. Conditioning on an independent string preserves the uniform product distribution of U, so the same argument applies at each supplied value. □

Define

$$b_j = \pi_j a_j, \qquad (v_1, v_2, v_P) = (b_1, b_2, r(b_1 + b_2)), \qquad v_* = \max(v_1, v_2, v_P). \tag{32}$$

Proposition 12 (Exact prediction optimum)**.** *Every encoder satisfies*

$$Q(Z) \leq h(v_1 x_1 + v_2 x_2 + v_P x_P) \leq h v_*.$$

The codes U_1, U_2, and $U_1 \oplus U_2$ attain hv_1, hv_2, hv_P, respectively. Thus

$$R^* = h(b_1 + b_2 - v_*), \qquad \Delta(Z) = hv_* - Q(Z).$$

Proof. Write $M_j = I(U_{j,S}; Z \mid S)$ and $N_j = I(U_{j,S}; Z \mid S, U_{3-j})$, with the request-$j$ mask distribution. Then $Q = \sum_j \pi_j[(1 - r)M_j + rN_j]$. Lemma 11 gives $M_j \leq a_j I_j$ and

$$N_1 \leq a_1(D - I_2), \qquad N_2 \leq a_2(D - I_1).$$

Substitution gives the displayed linear bound. All three displayed codes meet the alphabet budget. A raw code reveals exactly its requested masked bits; parity reveals none of either string alone and all requested bits when the other string is supplied. Their gains therefore equal the three displayed values. Finally, $H(X \mid V) = h(b_1 + b_2)$. □

H.2 Power objectives

For $t > 0$, set

$$C_j = (1 - a_j + a_j 2^t)^n, \qquad G_j = \pi_j(C_j - 1),$$
$$(w_1, w_2, w_P) = (G_1, G_2, r(G_1 + G_2)), \qquad \epsilon_n = \sum_j \frac{G_j}{e a_j h}. \tag{33}$$

Here $t = \alpha - 1$ and G_j is the weight denoted by w_j in (1); (w_1, w_2, w_P) lists the three allocation weights.

Theorem 13 (A uniform bound over the encoder class)**.** *For every randomized encoder within the alphabet budget,*

$$0 \le F_t(Z) - 1 \le w_1 x_1 + w_2 x_2 + w_P x_P + \epsilon_n. \tag{34}$$

Writing $w_* = \max(w_1, w_2, w_P)$, the attained witnesses imply

$$1 + w_* \le F_t^* := \max_Z F_t(Z) \le 1 + w_* + \epsilon_n. \tag{35}$$

If w_k is the unique largest weight, put

$$\Gamma = w_k - \max_{\ell \ne k} w_\ell > 0, \qquad \zeta = \min\{1, (\delta + \epsilon_n)/\Gamma\}.$$

Every code with $F_t(Z) \ge F_t^* - \delta$ satisfies $x_k \ge 1 - \zeta$. With $d_k = v_* - v_k$ and $c_k = \pi_k$ for $k = 1, 2$, or $c_P = r$,

$$h d_k (1 - \zeta) \le \Delta(Z) \le h d_k + c_k h \zeta. \tag{36}$$

Proof. Fix request j, a supplied value u of the other, independent string when present, and a nonempty mask of size s. The target is still uniform on 2^s values. Hence its information density $\iota = \log L$ is at most $s \log 2$. The exponential chord on $[0, s \log 2]$, and the sign of its left side for $\iota < 0$, give

$$e^{t\iota} - 1 \le \frac{2^{ts} - 1}{s \log 2}\, \iota_+.$$

Under the conditional joint distribution, $\mathbb{E}\iota_+ = I(U_{j,S}; Z \mid U_{3-j} = u) + \mathbb{E}\iota_-$. Changing to the conditional product measure gives

$$\mathbb{E}\iota_- = \mathbb{E}_{\text{product}}[L \log(1/L) \mathbf{1}_{\{L<1\}}] \le 1/e.$$

Thus the moment $A_{S,u}$ obeys

$$A_{S,u} - 1 \le \frac{2^{ts} - 1}{s \log 2} \left[I(U_{j,S}; Z \mid U_{3-j} = u) + \frac{1}{e} \right]. \tag{37}$$

At $s = 0$ its contribution is zero.

Average first over uniform size-s masks. Lemma 11 bounds their mean information by $(s/n) I(U_j; Z \mid U_{3-j} = u)$. For the random mask size T, the function $g(s) = (2^{ts} - 1)/s$ for $s \ge 1$, $g(0) = 0$, is increasing. Consequently

$$0 \le \operatorname{Cov}(T, g(T)) = C_j - 1 - n a_j \mathbb{E} g(T).$$

Binomial averaging of (37) therefore yields

$$A_{j,u} - 1 \le (C_j - 1) \left[\frac{I(U_j; Z \mid U_{3-j} = u)}{h} + \frac{1}{e a_j h} \right].$$

Average over the other string values. Apply the same argument without the other string, and then average requests and availability. Since $I(U_1; Z \mid U_2) = h(x_1 + x_P)$ and $I(U_2; Z \mid U_1) = h(x_2 + x_P)$, the information terms become $G_1(x_1 + r x_P) + G_2(x_2 + r x_P)$. The two availability weights sum to one for each request, giving exactly ϵ_n. Jensen gives $F_t \ge 1$.

The three witnesses have moments $1 + w_1, 1 + w_2, 1 + w_P$, proving (35). A maximum exists: stochastic encoder matrices form a compact product of simplices. At a fixed context, the contribution of an output of probability p_z is $2^{t|S|} p_z \sum_x P(x \mid z, v)^{1+t} \le 2^{t|S|} p_z$, so continuity extends to vanishing output marginals.

Set $w_0 = 0$. The witness lower bound and (34) give

$$\sum_{i \in \{0,1,2,P\}} (w_* - w_i) x_i \le \delta + \epsilon_n.$$

The gap Γ then implies $x_k \ge 1 - \zeta$. The prediction envelope yields $Q/h \le v_k x_k + v_*(1 - x_k)$, proving the lower regret bound. When the largest weight is w_k for $k \in \{1, 2\}$, $H(U_k \mid Z) \le h\zeta$. Lemma 11 gives $M_k \ge a_k h - h\zeta$. Supplying the other, independent string cannot decrease this mutual information, so $Q \ge h v_k - \pi_k h \zeta$. For $k = P$,

$$H(U_1 \mid Z, U_2) = h(x_0 + x_2) \le h\zeta, \qquad H(U_2 \mid Z, U_1) \le h\zeta.$$

The cases where the other string is supplied give $Q \ge h v_P - r h \zeta$. These prove the upper regret bound. □

The value bounds hold at ties, but the concentration claim requires a positive gap. They do not classify exact finite-size optimizers.

H.2.1 THE SCALE NEAR KL

For fixed $c > 0$ and $t = c/n$,

$$G_j \longrightarrow \gamma_j(c) = \pi_j(2^{a_j c} - 1), \qquad \epsilon_n = O(n^{-1}).$$

Away from a tie in $(\gamma_1, \gamma_2, r(\gamma_1 + \gamma_2))$, every sequence with moment gap $\delta_n = o(1)$ satisfies

$$\Delta(Z) = h(v_* - v_k) + o(n). \tag{38}$$

For exact optimizers the remainder is $O(1)$. A gap of $o(n)$ in the power divergence $(F_t - 1)/t$ is equivalent to this moment-gap condition. For a pooled Rényi gap η, the moment gap is $F_t^*(1 - e^{-t\eta})$; at this scaling F_t^* stays bounded, so $\eta = o(n)$ is equivalent as well.

Equal observation quality. If $a_1 = a_2 = a$, the power weights are $(C - 1)(\pi_1, \pi_2, r)$ and the prediction weights are $a(\pi_1, \pi_2, r)$. Let $g > 0$ be the gap between the largest entry of (π_1, π_2, r) and the next one. For every positive t and every exact optimizer,

$$0 \le \Delta(Z) \le \frac{c_k}{eag}. \tag{39}$$

Indeed $\epsilon_n/\Gamma = 1/(eahg)$ in Theorem 13. This bound is uniform in t, but not as a or g tends to zero. It concerns exact optimization: if $t_n = n^{-2}$, even a constant code has moment gap $O(n^{-1}) = o(1)$ while losing a linear amount of predictive information. Approximate optimization must resolve the actual power-weight gap.

Unequal observation quality. Take $\pi = (.75, .25)$, $a = (.5, .9)$, and $r = .9$. Prediction's weights are $(.375, .225, .54)$, so parity is optimal. The ratio

$$\frac{\gamma_2(c)}{\gamma_1(c)} = \frac{2^{\cdot 9c} - 1}{3(2^{\cdot 5c} - 1)}$$

increases strictly from .6 to infinity. Its logarithmic derivative is positive because $b/(1 - e^{-bc})$ increases with $b > 0$. Parity has the largest limiting weight until this ratio reaches $r/(1 - r) = 9$, at $c \approx 11.8292$. Above that crossing, every qualifying sequence satisfies

$$\Delta(Z) = .315n \log 2 + o(n).$$

Below it the regret is $o(n)$; exact optimizers have $O(1)$ regret there. These are asymptotic statements away from the crossing, not a finite-size optimizer switch. If $r = 1$, parity is an exact common optimum for prediction and every power criterion with $\alpha = 1 + t > 1$: it attains the pointwise moment cap at each mask.

H.3 PEARSON WHOLE-STRING RECOVERY

At $t = 1$, set $T(Z) = F_1(Z) - 1$ and $T^* = \max_Z T(Z)$, using the unhalved Pearson convention. The following result controls a stronger observable than information allocation: recovery of all n bits of one string from the stored code alone.

Theorem 14 (Whole-string recovery and predictive regret)**.** *Suppose $a_2 > a_1 > 0$ and $r < 1$. Put*

$$C_j = (1 + a_j)^n, \qquad q_2 = \frac{1 - a_2}{1 + a_2}.$$

If $T(Z) \ge T^* - \delta$, define

$$e_n = \min\left\{1 - 2^{-n}, \frac{\pi_1(C_1 - 1) + \delta}{\pi_2(1 - r)C_2(1 - q_2)}\right\},$$
$$B_n = h_2(e_n) + e_n \log(2^n - 1), \tag{40}$$

where h_2 is binary entropy in nats. There is a decoder of U_2 from Z alone with error at most e_n, and $H(U_2 \mid Z) \le B_n$. With $d = v_* - \pi_2 a_2 \ge 0$,

$$d(h - B_n) \le \Delta(Z) \le hd + \pi_2 B_n. \tag{41}$$

The same conclusions hold under the weaker achieved-score condition $T(Z) \ge T(U_2) - \delta$.

Proof. Let A_j be the order-two moment when the other string is absent, and $\bar{A}_j$ its conditional counterpart averaged over the supplied string. Uniformity of the target and data processing give $A_j, \bar{A}_j \le C_j$. Storing U_2 attains $T(U_2) = \pi_2(C_2 - 1)$. The achieved-score premise and the caps on all remaining modes give

$$\pi_2(1-r)(C_2 - A_2) \le \pi_1(C_1 - 1) + \delta. \tag{42}$$

This isolates the case where the second string is requested without its other string; its positive probability is essential.

Write R_z for the posterior of the entire U_2 given $Z = z$. Include the independent mask in the observed target $(S, U_{2,S})$. For raw strings u, u', the inner product of the two conditional-target likelihood ratios under this observation's marginal equals

$$(1+a_2)^{n-d_H(u,u')}(1-a_2)^{d_H(u,u')}.$$

Each agreeing coordinate contributes $1 + a_2$, and each disagreeing coordinate contributes $1 - a_2$. Expand the squared norm of the posterior mixture of these ratios and average Z to obtain

$$A_2 = C_2 \mathbb{E}_Z \mathbb{E}_{U,U' \text{ iid} \sim R_Z} q_2^{d_H(U,U')}. \tag{43}$$

The first string and encoder randomness are integrated into the channel from U_2 to Z; the posterior here does not condition on a supplied other string.

Using $1 - q_2^d \ge (1 - q_2)\mathbf{1}_{\{d>0\}}$ gives

$$\frac{C_2 - A_2}{C_2(1-q_2)} \ge \mathbb{E}_Z \left[1 - \sum_u R_Z(u)^2 \right] \ge \mathbb{E}_Z[1 - \max_u R_Z(u)].$$

The last quantity is the MAP error. Combine this with (42), clip by the trivial bound $1 - 2^{-n}$, and apply Fano's inequality. Its right side is increasing on this interval, proving (40) and $0 \le B_n \le h$. At $a_2 = 1$, interpret the zero-distance term 0^0 as one.

Let $b = H(U_2 \mid Z) \le B_n$. Independence and the code budget imply

$$I(U_1; Z) \le b, \qquad I(U_1; Z \mid U_2) = D - I_2 \le b.$$

Thus Lemma 11 gives $M_1, N_1 \le a_1 b$ and $a_2 h - b \le M_2 \le N_2 \le a_2 h$. In particular, $Q \ge \pi_2 a_2 h - \pi_2 b$, proving the upper regret bound. For the lower, $x_2 = 1 - b/h$ and Proposition 12 give

$$Q/h \le v_2 x_2 + v_*(1 - x_2), \qquad \Delta \ge d(h - b) \ge d(h - B_n).$$

The maximum exists by the continuity argument used above. None of these bounds restricts the encoder to copying coordinates or being deterministic. □

For fixed $r < 1$, positive request probabilities and $a_2 > a_1$, exact optimizers have a MAP-error upper bound $O(\kappa^n)$, where $\kappa = (1+a_1)/(1+a_2) < 1$. Then $B_n = O(n\kappa^n)$ and $\Delta = hd + O(n\kappa^n)$. More generally, $\delta_n = o(C_2)$ gives vanishing whole-string decoding error, $B_n = o(n)$, and $\Delta = hd + o(n)$. This is stronger than merely having an $O(1)$ conditional-entropy deficit: omitting one uniform bit leaves block-decoding error $1/2$.

For the same asymmetric example as Section H.2, $d = .315$. At $n = 128$, exact rational arithmetic gives

$$e_{128} = \frac{95}{3} \frac{(3/2)^{128} - 1}{(19/10)^{128}} < 3 \times 10^{-12}.$$

Using $h_2(e) \le e[\log(1/e) + 1]$, $\log 10 < 2.303$ and $\log 2 < .694$ then gives $B_n < 3.52404 \times 10^{-10}$ and, conservatively,

$$|\Delta(Z) - .315 \cdot 128 \log 2| < 1.33 \times 10^{-10} \text{ nats}.$$

The center is approximately 27.9477 nats. This evaluates a theorem about every exact optimizer; it is not a numerical search for one.

H.4 AVAILABILITY AND OPTIMIZATION ACCURACY

The factor $1 - r$ cannot be dropped when availability varies with n. For fixed request probabilities and retentions, sufficient recovery conditions are

$$\frac{C_1 - 1}{(1 - r_n)C_2} \longrightarrow 0, \qquad \frac{\delta_n}{(1 - r_n)C_2} \longrightarrow 0.$$

They do not constitute a sharp transition classification.

To see the role of the second condition, put $\mathcal{C}_n = \pi_1(C_1 - 1) + \pi_2(C_2 - 1)$. The cap gives $T^* \le \mathcal{C}_n$, while parity attains $r_n\mathcal{C}_n$. At $r_n = 1 - 1/n$, parity is within $\mathcal{C}_n/n = o(C_2)$ of optimal, yet its whole-string MAP error without the other string is $1 - 2^{-n}$. Exact optimizers instead satisfy the recovery bound $O(n\kappa^n)$. Vanishing relative objective error alone does not resolve a diminishing probability of requesting one string without receiving the other.

There is also a finite-size window where parity is exactly optimal below full availability. The Pearson objective is convex in the encoder kernel, since each conditional moment is a sum of quadratic-over-linear terms. A deterministic encoder therefore attains a maximum. For a fixed supplied value and a full mask, its conditional moment is the number of distinct outputs used by that conditional map. If any such map is not bijective, it loses at least one unit from the full-mask cap. Its deficit, averaged over the modes with the other string supplied, is consequently at least

$$g_n = 2^{-n} \min_j \{\pi_j a_j^n\} > 0.$$

Every such deterministic encoder satisfies $T \le r(\mathcal{C}_n - g_n) + (1 - r)\mathcal{C}_n$. An encoder bijective in each string for every fixed value of the other string has uniform outputs given either string, and thus $A_1 = A_2 = 1$; its value is $r\mathcal{C}_n$, as for parity. It follows that parity is exactly optimal whenever $(1 - r)\mathcal{C}_n < rg_n$. For example, $r_n = 2\mathcal{C}_n/(2\mathcal{C}_n + g_n)$ lies in this window. Convexity extends the deterministic comparison to the randomized class.

The finite-n optimum value is continuous in r: it is the maximum of finitely many affine deterministic-code values. The behavior is nonuniform as n grows and r approaches one; the finite value itself does not have a discontinuity.

Under the Pearson convention divided by two, an objective gap η enters Theorem 14 as $\delta = 2\eta$. For pooled order-two Rényi, gap η corresponds to $(1 + T^*)(1 - e^{-\eta})$. Thus $\eta_n = o(1)$ suffices at fixed r, and $\eta_n = o(1 - r_n)$ is a corresponding sufficient scale under the displayed varying-availability conditions. These differ from the near-KL $o(n)$ normalized-power tolerance.

H.5 NUMERICAL SCOPE

Small finite-distribution calculations check the power envelope, exact witnesses, conditional marginals, regret implications and the finite parity window. Their enumeration counts and limitations are recorded in `numerical_support/appendix_records/`. The arbitrary-size and every-encoder conclusions follow from the proofs above.

The numerical entries in Table 1 evaluate (36). They are floating-point evaluations of proved bounds; they are neither sampling confidence intervals nor rigorous numerical error enclosures, and do not come from a numerical search for optimal codes.

H.6 REQUEST-ONLY SPECIAL CASE

Setting $r = 0$ recovers the source in which only the request and mask arrive after encoding. Including S in the target instead of the context changes only its encoder-independent entropy. If $a_2 > a_1$ but $\pi_1 a_1 > \pi_2 a_2$, put $d = \pi_1 a_1 - \pi_2 a_2$. Prediction stores U_1, whereas Theorem 14 forces recovery of U_2 for sufficiently precise Pearson optimization. In its notation, $d(h - B_n) \le \Delta \le dh + \pi_2 B_n$. At $\pi = (3/4, 1/4)$, $a = (1/2, 9/10)$, $n = 128$ and exact optimization, the MAP-error bound is below 10^{-12} and $|\Delta - 19.2 \log 2| < 4.5 \times 10^{-11}$ nats.

Theorem 13 gives the request-only power weights $G_j = \pi_j[(1 - a_j + a_j 2^t)^n - 1]$ and the same finite allocation bounds. For $t = c/n$, their limiting crossing solves $\pi_1(2^{a_1 c} - 1) = \pi_2(2^{a_2 c} - 1)$, numerically $c = 2.9884644481$ for these parameters. The monotonic ratio argument above proves

uniqueness. Away from the crossing, moment gaps $o(1)$ give regret $o(n)$ below and $dn \log 2 + o(n)$ above; exact optima have $O(1)$ remainders. At $n = 128$, the weights of the two raw codes are equal at $\alpha = 1.0235322927$. Near this order, the bounds do not determine the exact optimum over all codes.

I LEARNING WITH A NORMALIZED TARGET

We now allow the training target to be learned, with identity covariance after averaging over requests and masks, and study the predictive information retained by the code. This controlled construction uses Appendix H.6's request-only source with unequal observation quality. Its source, teacher and optimizer differ from Section 3; it does not evaluate a modern self-supervised method or natural image/language data.

I.1 SOURCE, TARGET AND POPULATION COMPARISON

Let U_1, U_2 be independent uniform sign vectors of length $n = 128$. An encoder stores n hard signs before a request J is known. The probabilities are $\pi = (p, 1-p) = (.75, .25)$; independent masks retain coordinates with probabilities $a = (.5, .9)$. The prediction workload asks for the retained bits of the requested string. All risks below omit the independent mask entropy. The exact optimal risk is $R^* = (1-p)a_2 n \log 2 = 19.9626388001$ nats per request, attained by storing U_1. Storing U_2 instead has regret 13.3084258668 nats.

The teacher receives only the masked target values. For each coordinate it has nonnegative masses $q_{1i} + q_{2i} = 1$, gains $g_{ji} = \sqrt{q_{ji}/\pi_j}$, and target

$$T_{ji} = g_{ji} S_{ji} U_{ji} / \sqrt{a_j}.$$

The pooled target has zero mean and identity covariance: $\sum_j \pi_j \operatorname{Cov}(T_j \mid J = j) = I$. This is a normalization over requests and masks; its covariance need not be identity conditional on a particular request or mask.

The student sees only the code and request, and predicts $\widehat{T}_{ji} = g_{ji}\sqrt{a_j}\, m_{ji}(Z)$, where $m_{ji}(Z) = \tanh(l_{ji}(Z)/2)$. It receives no mask or unthresholded encoder output. The teacher gains appear on both sides of the squared error and are jointly optimized; there is no moving-average teacher or stopped target gradient. Integrating the independent masks gives the total MSE

$$L = \sum_{j,i} q_{ji} \left\{ 1 - a_j + a_j \mathbb{E}\left[(U_{ji} - m_{ji}(Z))^2 \right] \right\}. \tag{44}$$

Every term is a convex combination of quantities at least $1 - a_2$, so $L \geq n(1 - a_2)$. Storing U_2, taking $q_{2i} = 1$, and predicting its bits exactly attains the bound. Since $a_2 > a_1$, equality requires $q_{1i} = 0$ and zero error on every bit of U_2. Thus $H(U_2 \mid Z) = 0$. The n-bit budget then forces Z to be a bijective relabeling of U_2 and independent of U_1, even if the allowed encoder is nonlinear or randomized. Every optimum in this closed teacher family therefore has predictive regret

$$\Delta = \{p a_1 - (1-p) a_2\} n \log 2 = 13.3084258668 \text{ nats}.$$

These are two consequences of the same code, not a conversion of MSE into log-loss units. A code depending only on U_1 and independent encoder randomness has best possible pooled MSE $L/n = 1 - a_1 = .5$.

The implementation uses $q_{ji} = \operatorname{softmax}_j(\theta_{ji})$. Its finite softmax and tanh parameters can approach the boundary, so .1 is an infimum rather than an attained minimum in that parameterization. A same-architecture U_2 encoder with bit predictor logits ± 20 and teacher log masses $(-12, +12)$ attains $.100000000034$ per coordinate. The within-request control fixes $g_{ji} = 1$; its target covariance is identity within each request after averaging masks. Its MSEs for the codes storing U_1 and U_2 are $.625$ and $.775$. The first is a feasible comparator, not a claimed optimum over arbitrary nonlinear codes. Direct NLL uses the same bit logits and task masks; its exact prediction optimum stores U_1.

I.2 FITTING, PREDICTORS AND THE PRECEDING FAILED COMPARISON

The encoder receives a fixed rotation of the 256 source signs and uses a linear path plus a width-512 GELU residual followed by hard signs. Its predictor has the same two-branch form. The twelve fits cross the three objectives, initial codes U_1 and U_2, and seeds 730201/730202. Each completes 10,240 updates of 1,024 raw inputs, evaluating both masked requests with their workload weights. Pooled teacher masses are fixed at $(.5, .5)$ for 2,048 updates and learned for the remaining 8,192. The final scheduled checkpoint is used without evaluation-row selection. Initial logits have magnitude two for the initially stored string. AdamW uses the moments, offset and clipping of Appendix B, with weight decay 10^{-4} except on the teacher. Its cosine schedule spans 10,240 updates. The full settings are in `numerical_support/appendix_records/`.

The preceding twelve-fit comparison used an identity straight-through derivative. Its likelihood controls also destroyed information in the initially optimal code, and every objective missed a feasible solution. Diagnostics found large growth of pre-sign magnitudes and a nonzero surrogate derivative along positive rescalings that leave every hard code unchanged.

The present comparison uses the complete RMS-normalized backward map defined in Appendix C.5, with the same hard forward code. Architecture, initialization, training streams, optimizer and update count match the preceding comparison. Validation and evaluation use fresh streams after those earlier results were inspected.

Encoders are trained in FP32, then weights and the input rotation are exported to FP64. Every predictor receives the literal 16-byte packed and unpacked code from that exported encoder. A fresh width-512 predictor trains for 4,096 updates as a product density, then 4,096 more updates with a causal residual. The learning-rate schedule restarts in each 4,096-update phase. The latter retains the product predictor as an exact submodel and can use only preceding revealed target bits. Predictor training balances the two requests; the first phase uses complete string labels, and the second equally mixes full and task masks. Full labels are additional predictor-training supervision, supplied equally to all objectives. They are not extra information available at prediction. For each request and mask regime, 4,096 separate validation rows choose among the fresh product predictor, its autoregressive continuation and the co-trained product predictor. These choices are fixed before evaluation on 65,536 fresh samples.

I.3 WHAT THE CONFIDENCE BOUNDS ESTABLISH

Let $I_j = I(U_j; Z)$ and $M_j = I(U_{j,S}; Z \mid S)$ for the task's specified mask distribution. Mixing a predictor's normalized joint density with the uniform density gives a lower bound on retained information:

$$q_\tau = (1-\tau)q + \tau 2^{-|S|}, \qquad G = \log q_\tau + |S| \log 2, \qquad \tau = .01.$$

Its mean is at most the retained information, and $\log \tau \le G \le n \log 2$. The source inequalities $M_j \le a_j I_j$ and $I_1 + I_2 \le n \log 2$ give, for simultaneous lower bounds ℓ on these four information quantities,

$$\max\{0, .15\ell_{I_2}, \ell_{M_2}/6\} \le \Delta \le .375 n \log 2 - .75\ell_{M_1} - .25\ell_{M_2}. \tag{45}$$

The lower side holds for every predictor of the code. The upper side bounds its predictive regret using achieved densities; it does not assume that a fitted predictor is optimal. A poor predictor is never used as evidence of low information.

The event covers 72 one-sided statements: four information lower bounds and two objective bounds for each of twelve fixed endpoints. With $N = 65,536$, each information radius is $(n \log 2 - \log \tau)\sqrt{\log(72/.05)/(2N)} = .6951777608$ nats. We clip information lower bounds at zero and their known entropy caps. MSE is analytically averaged over masks, with range width bounded by $1 + 3a_2 = 3.7$. The reported NLL objective uses the co-trained predictor's distribution with the same mixture over the complete masked target; its range width, after division by n, is at most $\log 2 - \log \tau / n$. Multiplying those widths by the same square-root factor gives the objective radii. The uncertainty unit is an independent raw source row. A union bound gives simultaneous 95% coverage without requiring independence across endpoints or requests. It conditions on fitted and validation-selected models, rather than on a population of possible training runs. Calculations use

FP64 arithmetic, not directed rounding. This event is separate from both the preceding identity-gradient experiment and the 144-statement matching-code event in Figure 2. We do not assign 95% coverage to their intersection.

I.4 All endpoints and their limits

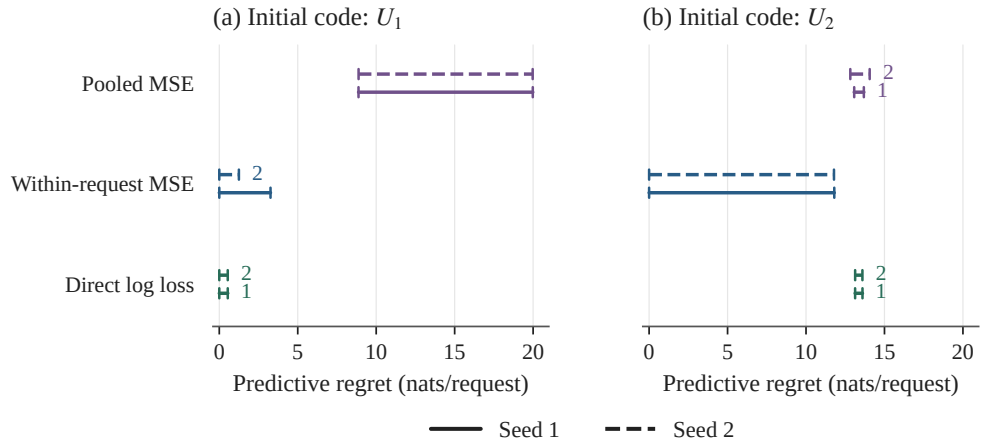

Figure 5: **Target normalization changes what the learned code retains.** Bounds on the minimum predictive regret of twelve codes: three objectives, two starts and two seeds. Solid and dashed brackets distinguish seeds; short brackets also carry the seed number. The bounds hold together with 95% confidence over fresh source rows, conditional on the fitted and validation-selected models; this event is separate from Figure 2. Pooled normalization loses predictive information from the optimal start U_1. The controls perform better from that start, but direct prediction from the U_2 start remains near that code's 13.31-nat regret. The planned five-nat advantage of both controls is established only from U_1. No pooled fit reaches its objective infimum.

All twelve code-regret intervals appear in Figure 5. Their exact endpoints and attained-objective intervals are in `numerical_support/normalized_target/endpoints.csv`, under the same 72-statement event. Objective values are pooled MSE, within-request MSE or smoothed NLL divided by 128, with different units.

On the same 95% event, both pooled-normalization seeds from the initial code U_1 beat the .5 optimum for codes based only on U_1. Their full-string information bound $\ell_{I_2} \geq 59.15283866$ nats gives $\Delta \geq .15\ell_{I_2} > 8.87$ nats for every predictor and $I_2/\log 2 \geq 85.33$ bits. From that start, within-request normalization and NLL keep predictive regret below 3.27 and .540 nats, separating both controls from pooled normalization by more than five nats in each seed. Since $I_1 + I_2 \leq 128 \log 2$, the pooled codes retain at most 42.67 bits about U_1, initially stored in full. These information bounds do not identify specific retained coordinates.

The prospective criterion required the five-nat separation in every seed *and both starts*; it is not met. From the initial code U_2, direct NLL remains near that string's 13.31-nat regret. The within-request control also has broad regret intervals from that start. All pooled-MSE objective lower bounds exceed the feasible value .100000000034, so none of the learned pairs attains the objective infimum. The within-request-MSE pairs likewise retain objective gaps to the comparator that stores U_1 in some cells. The good-start result does not establish reliable convergence.

Retrospectively, disjoint held-out objective intervals favor the U_2 start for pooled MSE and U_1 for both controls, in both seeds. These comparisons use the same event but do not validate a prospective selection rule. Their exact values are preserved in `numerical_support/appendix_records/SELECTION_DETAILS.md`.

Sources and verification scope. The numerical-support README and `numerical_support/normalized_target/SOURCES.json` identify the supplied summaries, source hashes and

verification records. All twelve saved row arrays were reaggregated. Independent network replay covered 1,024 rows of one endpoint, not every forward or the training process. The accompanying methods record enumerates the static input, normalization, serialization and gradient checks.

J EXACT CONDITIONAL MATCHING FOR THE BINARY SOURCE

We use the conditional contrastive decomposition developed by Sordoni et al. (2021). The statement below gives exact agreement on the finite-bit source of Appendix A.

In the target-candidate orientation, condition on the request, mask, supplied opposite string and preceding requested bits. The next selected bit is marginally uniform before the code is observed.

Proposition 15 (Binary conditional matching and predictive log loss)**.** *For each selected target bit, draw one independent fair negative bit and randomize the positive position among the two candidates. Let the critic observe the code, the supplied context and the preceding target bits. For every admissible encoder, the sum of the Bayes matching gains is*

$$\mathcal{J}_{\mathrm{chain}}^{\mathrm{target}}(Z) = \tfrac{1}{2} I(X; Z \mid V).$$

Consequently the maximizing code sets agree with prediction, and a code within ϵ of the Bayes chain optimum has predictive regret at most 2ϵ.

More generally, take any strictly positive normalized autoregressive predictor $q(x \mid z, v) = \prod_l q_l(x_l \mid z, v, x_{<l})$ and define its candidate probabilities by normalizing q_l over the two candidate values. Its summed matching loss satisfies

$$L_{\mathrm{chain}}(q, Z) = \tfrac{1}{2}\mathbb{E}|S| \log 2 + \tfrac{1}{2} L_{\mathrm{prediction}}(q, Z).$$

Thus the two losses have identical minimizers over any shared class of encoder–predictor pairs under this scoring rule. A finite shared per-candidate score already has this form: normalizing its exponentials over the two possible bit values defines q_l.

Proof. Fix a selected bit and write $A = (V, X_{<l})$. The negative is a fair coin independent of (X_l, Z, A). With probability one half it equals the positive, so the two candidate probabilities are both $1/2$. Otherwise the candidate values are zero and one. Their predictor probabilities sum to one, and the matching loss is $-\log q_l(X_l \mid Z, A)$. Both events have probability one half independently of (X_l, Z, A). The expected loss for this bit is therefore $\frac{1}{2}\log 2 + \frac{1}{2}\mathbb{E}[-\log q_l(X_l \mid Z, A)]$. Sum over the selected bits, including zero contribution for an empty mask. For the Bayes critic, on unequal candidates the positive-index posterior is the posterior probability of its bit. This remains well-defined on the sampling support when that posterior has zeros. Replace predictive cross entropy by conditional entropy and use the chain rule. Since the selected bits are conditionally uniform before Z is seen, $H(X \mid V) = \mathbb{E}|S| \log 2$. Subtracting matching loss from $\mathbb{E}|S| \log 2$ gives the gain identity. The regret statement follows by taking suprema, whether or not attained. □

This binary specialization requires fair conditional target marginals, target candidates and a shared encoder–predictor class. It need not hold for nonuniform marginals, a separately chosen predictor or code candidates. Comparisons grow with requested length; whole-target two-candidate matching has no such guarantee. Training and likelihood evaluation use target prefixes, whereas generation uses generated prefixes. The encoder sees none of this later information. Restricted predictors need not attain Bayes risk, and neither identity guarantees training convergence.

K NATURAL-DATA COMPARISONS

These studies ask whether the controlled distinction between stored information and predictor fit appears with natural image features and annotations. For small finite codes, the best predictor of each code can be computed exactly under the finite evaluation distribution, separating the code comparison from predictor fitting (Appendices K.1 and K.2). For larger codes, the predictive comparisons use fitted predictors (Appendices K.3 and K.4). None of these studies establishes a natural-population penalty.

K.1 PROFILING EVERY PREDICTOR OF A FINITE CODE

Suppose an encoder stores one symbol Z from a small alphabet for each image, possibly drawn at random from an encoder channel. A later request A, with probability w_a, asks for a binary or categorical target Y_a, and the predictor sees only (Z, A). Under the finite distribution of evaluated images and their requests, let m_z be the probability of symbol z and t_{za} the conditional distribution of Y_a given $Z = z$. These quantities integrate over the channel: the symbol is redrawn for each evaluated image rather than sampled once as a fixed archive. The minimum expected log loss over every predictor of the code is

$$R = \sum_a w_a \sum_z m_z \, H(t_{za}),$$

attained by the table t itself. We call this minimization over predictors *profiling*. A table θ fitted on training rows is an attainable predictor, with loss

$$L(\theta) = R + \sum_a w_a \sum_z m_z \, \mathrm{KL}(t_{za} \,\|\, \theta_{za}) \geq R.$$

In code-candidate matching, K candidate codes are drawn independently from m, the target is generated from a uniformly placed candidate, and the critic identifies that candidate. The Bayes critic's gain is

$$J_K = \sum_a w_a \, \mathbb{E}_{Z_1,\ldots,Z_K} H\Big(\tfrac{1}{K} \textstyle\sum_{k=1}^{K} t_{Z_k a}\Big) - R \in [0, \log K].$$

For a one-bit code the candidate mixture depends only on the number $B \sim \mathrm{Bin}(K, m_1)$ of candidates equal to one, so a binomial sum evaluates J_K exactly.

For a matching-trained code C and a direct-trained code D, a positive attained margin $R(C) - L(\theta_D)$ would show that the fitted direct table beats every predictor of C on this finite distribution. The profiled difference $R(C) - R(D)$ orders only the two codes' best predictors; the margin is that difference minus the direct table's fitting excess $L(\theta_D) - R(D)$. These are exact statements about a finite empirical law, evaluated in floating point. They are not population guarantees.

K.2 FINITE CODES ON DEEPFASHION AND CUB

A quantized supervised statistic. The first DeepFashion (Liu et al., 2016) study selects codes from a prescribed finite family rather than training an image encoder. It quantizes two TRAIN-fitted logistic probabilities u for denim and floral, computed from pinned CLIP (Radford et al., 2021) ViT-B/16 features; objectives use the actual annotations under the workload $(.725, .275)$, fixed earlier. One-bit codes threshold 720 prescribed linear directions of u at every distinct TRAIN cut, including axes and constants, without splitting ties, and store $Z = 1$ with probability $10^{-6} + (1 - 2 \times 10^{-6})h$ for threshold indicator h. Two-bit codes embed this family and add 66,049 TRAIN-quantile rectangles. The TRAIN search evaluates direct NLL and J_2 at every cut; J_{16} and J_{128} use the binomial sum or exclude cuts by a Jensen bound below an incumbent. All six selected functions and their TRAIN predictors are fixed before TEST, and J_{128} and direct select literally the same one-bit function. The float64 search has no rigorous numerical enclosure and implies no continuous-angle or population optimality. Dataset roles, preprocessing, angles, thresholds and tolerances are in `numerical_support/appendix_records/SELECTION_DETAILS.md` and the methods record.

On held-out TEST moments, the one-bit J_2-selected code has profiled risk .00677914 nats above the direct-selected code (paired 95% interval $[.00304291, .01064486]$), falling to .00072733 $[-.00114376, .00250721]$ at two bits. The TRAIN tables' excess over the profiled risk, .000050–.000061 nats for these one-bit codes, explains little of the .00676824 frozen-predictor gap. More directly, $R(C) = .1786415521$ and $L(\theta_D) = .1719235649$ imply an attained advantage of .0067179871 nats/request over every predictor of the matching code and request on this finite law, evaluated in floating point. The profiled-risk interval above is not an interval for this attained margin. Intervals use 4,000 paired filename-family bootstrap draws (seed 213259) that recompute nonlinear moments, conditional on the fixed functions and an empirical distribution weighting image rows equally, without refitting. The primary evaluation set excludes 404 SHA256 matches to previously used images, leaving 27,221 images in 5,403 filename-parent groups, all represented in TRAIN. The full 27,625-row TEST set is a prespecified secondary comparison. These evaluations do

not establish generalization to new filename families, disjointness from pretraining or independent garment identities. Checks using duplicate groups and the full evaluation set preserve the qualitative risk result.

The J_2-selected code's profiled J_2 exceeds the direct-selected code's by .00129823, with realized interval $[.00003880, .00248234]$, but its lower quantile is unresolved at Monte Carlo precision: 83 saved draws are nonpositive and the order-statistic band for the 2.5% quantile is approximately $[-.0000044161, .000078142]$. No draws or seeds were changed. Opposite population orderings of the two objectives therefore remain unconfirmed. The numerical record gives all frozen and profiled losses.

Learned categorical codes. Two later studies train small categorical encoders on frozen image features. On DeepFashion, the input is the frozen 512-dimensional CLIP ViT-B/16 feature of each image in the source TRAIN role of the DeepFashion annotation metadata released by Ma et al. (2020). The computational supplement's `provenance/MODEL_PROVENANCE.md` identifies this metadata and the pinned CLIP implementation. Splitting by filename parent gives 165,726 fitting, 21,700 calibration and 33,577 evaluation rows (4,225, 560 and 832 groups); the evaluation rows had been used in earlier analyses. A request asks, uniformly at random, for one of the 1,000 binary annotations. On CUB (Wah et al., 2011), frozen 768-dimensional SigLIP 2 image features (Tschannen et al., 2025) are used for images with exactly one positive bill-length class (of three) and one breast-pattern class (of four), every rating in these fields being at least 'probably' certain: 4,094 fitting, 833 calibration and 4,784 previously exposed test images. A request asks for bill length with probability .75 and breast pattern with probability .25. This separate-request workload was adopted after an exact calculation over deterministic codes of the true labels found the two objectives in agreement for joint twelve-class prediction, and before any image-encoder fit; joint prediction is a prespecified control at the primary capacity. The 3:1 weights are a declared workload, not a measured question frequency.

Each encoder maps standardized features to the floored channel $(1 - 10^{-4})\,\mathrm{softmax}(g) + 10^{-4}/M$ over M symbols and stores one draw per image; the predictor receives only that symbol and the later request. DeepFashion uses MLPs with 64 GELU units and $M = 8$ (primary) or $M = 2$, and a linear map with $M = 8$. CUB uses MLPs with $M = 4$ (primary) or $M = 8$, and a linear map with $M = 4$. For each encoder and seed, a warm fit first minimizes profiled risk on the fitting rows (joint risk for CUB). Matching (J_2) and direct (profiled risk) continuations then start from the same calibration-selected warm weights with reset Adam states. Every fit completes 200 full-batch updates of the exact objective computed from fitting-row moments; calibration selects a checkpoint from steps $0, 25, \ldots, 200$, including step zero, and both selected and final states are evaluated. DeepFashion's fitted predictor is the unsmoothed fitting-row table. CUB's mixes the joint fitting table with 1% uniform mass over the twelve classes, and its field predictors marginalize that table. There are 27 DeepFashion fits (seeds 924101–924103) and 33 CUB fits (seeds 924301–924303). CUB quantities integrate the channel exactly; no packed-symbol deployment was executed.

Table 9: All 42 selected and final comparisons of matching and direct continuations, under each study's training workload. Columns count pairs by the matching-minus-direct change in J_2 and profiled risk R: better on both (J_2 up, R down), tradeoff (both up), identical code, or worse on both (J_2 down, R up). The attained margin $R(C) - L(\theta_D)$ is in nats per request. Ranges span the displayed pairs, not confidence intervals, and are rounded outward.

Study	Encoder (M)	Both better	Tradeoff	Same code	Both worse	Attained margin
DeepFashion	MLP (8)	6	0	0	0	$[-.000575, -.000545]$
DeepFashion	MLP (2)	6	0	0	0	$[-.000308, -.000302]$
DeepFashion	Linear (8)	2	4	0	0	$[-.000517, -.000500]$
CUB request	MLP (4)	2	0	2	2	$[-.0865, -.0097]$
CUB joint	MLP (4)	2	0	2	2	$[-.1675, -.0242]$
CUB request	MLP (8)	1	2	2	1	$[-.1534, -.0508]$
CUB request	Linear (4)	3	2	0	1	$[-.0294, -.0110]$

No fitted direct table beats every predictor of its paired matching code: all 18 DeepFashion and all 24 CUB attained margins are negative (Table 9). The direct table's fitting excess, about .0003–.0006 nats on DeepFashion and .0097–.170 nats on CUB, exceeds every profiled cost of the matching code.

On DeepFashion, both MLP encoders improve J_2 and profiled risk under matching, relative to direct continuation, at every selected and final endpoint; all matching and direct endpoints improve their own objectives over the selected warm state. The linear encoder has small profiled tradeoffs in two seeds. A better profiled risk does not consistently improve the matching code's own fitted-table loss. Pointwise filename-family bootstrap ranges (4,000 draws) are descriptive only: re-profiling the matching table within each resample shifts every range below its original attained margin, so they are not confidence intervals.

On CUB, twelve continuations select step zero, so six selected matching/direct pairs share the identical warm code. All continuations still complete 200 updates; step zero has the strictly best calibration score in these cases. Four comparison pairs show profiled tradeoffs with a matching gain over warm; the linear tradeoffs appear only at the final checkpoints. Every final state improves its own objective on the fitting rows, but 16 of 33 final calibration objectives worsen, and seven matching endpoints lose test J_2 relative to warm. The first two primary seeds' final matching states are worse than direct in both objectives. Exact checkpoint choices and per-seed contrasts are preserved in `numerical_support/appendix_records/selection_consolidation/`.

`numerical_support/natural_categorical/` retains numerical records for all 54 DeepFashion and 66 CUB states, all 42 pairs, both CUB workloads and field-level classification summaries. These studies average over a fresh symbol per image and use previously exposed evaluation frames. They establish neither natural-population information loss nor objective agreement, and natural image inputs do not make the uniform annotation request or the 3:1 field request a natural workload.

K.3 All DeepFashion annotations with fitted predictors

A separate DeepFashion study uses all 1,000 annotation bits and frozen 512-dimensional CLIP features on the same source TRAIN role, with the same filename-parent split as the categorical study. These rows were used earlier for aggregate entropy inspection and serialization checks; the evaluation rows are held out from this study's fitting and selection, but not from those earlier analyses. Protected TEST is unused. Preprocessing, PCA, priors and code normalization use fitting rows only.

Joint matching, five-group matching and direct NLL each train a separate PCA-initialized residual 512-to-128 encoder with real-valued codes and a 128–512–1000 predictor. Matching uses product-Bernoulli scores with one denominator over all labels or one per attribute group. Direct summed marginal BCE is proper for marginals, not the true joint annotation likelihood. A learned linear code, fixed PCA128 and the full 512-dimensional input are controls. All four trained encoders complete 8,192 updates; validation every 512 updates selects their checkpoints. Each of the six codes receives two logistic predictors and a nonlinear residual predictor initialized from the better logistic fit, all with 8,192 updates. Validation selects among these and the co-trained head before evaluation. The methods record gives the 22-fit design and optimizer settings, and `numerical_support/appendix_records/deepfashion_all_annotations.csv` the complete BCE and macro-AP endpoints. BCE sums 1,000 marginal losses; AP averages over supported labels.

Joint-minus-direct BCE falls from 1.222901 to .007393 (about 99.4% removed), with paired 95% interval $[-.006391, .021072]$. The row-weighted bootstrap resamples 832 filename families, conditional on fixed policies, without refitting variation. The five-group, joint-versus-five-group and learned-linear comparisons are also unresolved; their exact contrasts are in `numerical_support/appendix_records/SELECTION_DETAILS.md`. PCA128's excess over nonlinear direct is .107965 $[.094008, .123085]$, while learned linear essentially matches direct. Full input has worse fitted BCE and higher AP despite permitting every compressed-code predictor; its fitted head need not attain that best possible risk. Both matching objectives and direct prediction select early checkpoints; later validation BCE worsens. The methods record retains exact checkpoint choices, predictor selections and exploratory candidate-score measurements. Those measurements neither establish nor exclude saturation and supply no natural-source information upper bound.

K.4 COMPACT OCR-VQA CODES: BOTH SEEDS AND PREDICTOR CONTROLS

The predetermined OCR-VQA training subset supplies title, author and genre requests (Mishra et al., 2019). A spatially attentive encoder maps frozen 196-by-768 SigLIP 2 tokens (Tschannen et al., 2025) to 32 latent slots, with four or sixteen hard-sign channels each: exactly 128 or 512 stored bits (16 or 64 bytes), before the request. Tokens, floating residuals and per-image identifiers cannot bypass the code. Both objectives continue a predictive warm code for the full 50-epoch schedule; calibration selects checkpoints, including the eligible starting encoder, and fresh four-/six-layer sequence predictors.

Both seeds complete 17 prediction endpoints and 54 candidate cells on 4,092 distinct image-hash groups: 4,092 titles, 3,814 authors and 4,092 genres (11,998 questions; 278 absent authors excluded). Sequence NLL includes EOS, with training request weights approximately $(.341235, .317531, .341235)$ and mean lengths $(13.84, 4.81, 4.22)$ tokens. Forwards use BF16; centered vocabulary normalization and sequence sums use FP64. Checkpoint and predictor choices use complete calibration evaluations; reserved evaluation losses do not enter either choice.

Prediction intervals below use 2,000 paired image-hash cluster bootstrap draws, preserving question multiplicity and request weights. They are pointwise descriptive 95% intervals conditional on all training, calibration and mixture selection, without multiplicity or refitting variation. Candidate intervals instead concern independent candidate sets conditional on the empirical evaluation bank. Both seeds share the reserved images and declared candidate-sampling and bootstrap seeds; they are not independent source replications.

Table 10: Both original OCR-VQA seeds, in weighted sequence NLL nats per request. Intervals are pointwise paired image-hash bootstrap intervals conditional on the selected models. Calibration selects a four-layer fresh predictor except for second-seed 128-bit matching (six layers). The second-seed 512-bit equality is literal selected network equality, not an equivalence test. Each gap is the selected fresh matching-encoder loss minus the selected fresh NLL-encoder loss.

		Co-trained		Selected fresh	
Seed	Bits	NLL encoder	Matching encoder	NLL encoder	Matching encoder
1	128	33.02387	42.90972	32.76631	32.77380
1	512	29.86779	52.30676	28.74640	29.94297
2	128	32.99124	40.43666	32.82225	32.70448
2	512	28.69427	28.69427	27.33724	27.33724
Selected fresh predictors		Gap [paired 95% interval]			
1	128	$.00750\ [-.03450, .04983]$			
1	512	$1.19657\ [1.12235, 1.26885]$			
2	128	$-.11777\ [-.16121, -.07695]$			
2	512	$0\ [0, 0]$			

Prediction, initialization and the adverse budget comparison. In seed one, refitting removes about 99.9%/94.7% of the between-encoder gaps at 128/512 bits. The larger budget improves both absolute risks but improves direct more: the prespecified $\mathrm{gap}_{128} - \mathrm{gap}_{512}$ is -1.18907 $[-1.27123, -1.10684]$, contrary to the hypothesis that a smaller budget amplifies the gap. At 512 bits matching hurts title prediction and helps author and genre. Six-layer predictors preserve this residual but do not exhaust predictor classes. Direct continuation selects its warm-start encoders; matching selects changed codes. Exact task and six-layer gaps are in the "OCR-VQA: first-seed task and predictor details" section of `numerical_support/appendix_records/METHODS_AND_CHECKS.md`. These are fitted-risk comparisons, not attained optima or information decompositions.

Seed two does not replicate the 512-bit penalty. Both objectives select their starting 512-bit encoders and identical four-layer predictors; every request gap is zero. At 128 bits NLL retains its start while matching changes its encoder, and refitting reverses their comparison in favor of matching. The

budget contrast is $-.11777$ $[-.16121, -.07695]$, again opposite the forecast. The numerical record retains all per-request values and predictor changes. Two fitted seeds do not characterize training variation.

Matching and source controls. The first seed's co-trained $K = 8$ matching-gain contrasts are unresolved; post-result comparisons using fresh predictors change the predictor as well as the code. Title prediction loss is not shown to purchase better title matching. Full-token and context-only controls show usable visual signal, but have different inputs and are neither Bayes oracles nor equal-budget competitors. In seed two, the selected 128-bit matching policy improves both prediction and $K = 8$ matching; its $K = 128$ matching contrast is unresolved. The 512-bit between-objective contrasts are zero. Although title and author have information lower bounds above $\log 8$ on the original simultaneous 54-cell event, this is under the empirical evaluation bank's candidate law; it neither separates objectives nor establishes natural-population information loss. The candidate comparisons and source-control values are retained in the numerical record.

A secondary forecast, fixed after seed one but before seed two was evaluated, concerned the 512-bit codes. It predicted positive matching-minus-NLL prediction gaps for title and request-weighted losses, and a decrease in the corresponding matching-gain contrasts when the candidate count rises from $K = 8$ to 128. All these contrasts are zero in seed two. At 128 stored bits, the request-weighted raw matching-minus-NLL gain changes by $-.04424$ from $K = 8$ to 128. That result does not establish the forecast made for the 512-bit codes. Neither this candidate-count forecast nor the primary code-budget hypothesis is therefore supported.

Predictor ensembles. Exploratory first-seed stacking (Yao et al., 2018) fits an answer-independent mixture of three complete predictive densities on calibration data, shared across requests. It improves prediction with up to three evaluations and no extra stored bits, without establishing an information ordering. The methods record gives its adverse task pattern and prior data exposure; this post-hoc analysis does not test the earlier forecasts.