MARSEA: MULTIVARIATE RELATIONS BY SELECTIVE EXCLUSION IN ATTENTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Attention represents relations pairwise: A_{ij} is a statement about one query and one key. Under **multivariate competitive exclusion** a set of candidates jointly bears on a target and competes for a finite influence budget, settled by comparison *within* the set: the winner takes the bulk, close rivals less, everything past a content-decided cutoff exactly none.

We prove four things about this class. **Separation**: no *query-local* scheme — rows computed independently per query, as in softmax with any positional bias or element-wise gate — can enforce a non-product-form constraint on a key's degree. **Conservation**: no row-normalized scheme can bound every key's fan-out below n_q/n_k, and doubly stochastic attention meets that bound only by giving every key the same content-free fan-out. **A precision identity**: with m gold items among n candidates, every full-support attention has evidence precision exactly m/n, whatever the scores. **Sharp recovery**: a sparsemax program over a relation slice returns the gold support exactly *iff* its exclusivity τ lies in $[1/(W + m\delta),\ 1/W)$, for score margin δ and within-set spread W; it degrades weakest-first above the interval, and the interval only widens as the slice shrinks.

MarSea is the mechanism these results leave room for. On the query or the key margin it places a bounded-size relation and, at each constrained element, one closed-form program that inherits the base normalizer's mass on the relation and re-shapes it by sparsemax, so members can receive *exactly* zero, under a shape control τ and a mass control λ. The query-local row form can learn a partition but not enforce one; the query-coupled column form can, and under causal decoding its exclusions are permanent. We deploy the row form on a softmax-one base and ablate the column form. On multi-value retrieval with a 1.5B decoder (three seeds), MarSea matches or exceeds softmax, softmax-one, causal learned-marginal transport and row-wise sparsemax on exact-set accuracy, precision and recall at every list length, with the fewest repeated and omitted values on 32-value lists; it beats softmax's exact-set accuracy by 6 and 13 points at 16 and 32 values, its margin over softmax-one alone is within seed variance, and row-wise sparsity alone hurts.

1 INTRODUCTION

An attention matrix is a weighted bipartite graph: A_{ij} is an edge from key j to query i. Softmax decides which edges carry mass *pairwise* — each weight from one query and one key — under a single constraint per row, $\sum_j A_{ij} = 1$. The relation it represents is *many-to-many*, and its only structure is the row budget.

Many relations are not of this form. Our object is **multivariate competitive exclusion**: a set $\mathcal{A}$ jointly bears on b, its members compete for a finite influence budget, and which prevails is settled by comparison *within* $\mathcal{A}$. Losing is graded near the top of the order and absolute below it — *little to nothing*, not uniformly nothing. We say *exclusion* rather than *prioritization*: row-softmax already ranks; what it cannot do is *end* the order at a cutoff the candidates themselves decide. The competing set may sit on either side of the graph — a relation about the *queries* is a statement about a *row*, the same relation about the *keys* a statement about a *column* — and the number that survives the cutoff

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.

need not be one: one-to-one, one-to-many and many-to-many exclusive relations are one object read from different margins.

The case we demonstrate on. At an intersection many signals may be present at once, under a priority order that differs by jurisdiction and so must be learned from content; each signal governs a *set* of lanes, and each lane is governed, in the end, by *one* signal. That is the **one-to-many** case: every element on one side has exactly one partner on the other, which may have many. It is the shape of retrieval — an answer token copies from one source span while a source span may be copied many times — and it is the case this paper instantiates, because a decoder's retrieval tasks present it and its partition gives every measurement a ground truth. Nothing in the mechanism or the theory is specific to it.

MarSea. We read A as a relational graph and place, on a constrained margin, a relation of bounded size per element — k candidates, chosen by a learned logit anchored on the attention score and supervised by block gold where the training data provide it — and at each such element a **content-dependent degree constraint**: one joint program over the element's slice that *inherits* the mass the base normalizer allocates to the slice and *re-shapes* it at an exclusivity τ, producing exact zeros, with a second control λ deciding how much mass may move between the slice and its complement. The program is sparsemax (Martins & Astudillo, 2016; Peters et al., 2019) — exactly solvable by sorting, non-iterative, unlike Sinkhorn attention (Sander et al., 2022) or capsule routing (Sabour et al., 2017). On the query margin this is the **row form**; on the key margin, the **column form**; a task selects one. The row form is itself query-local: it can *learn* a partition and *select* within a row, but by the separation theorem it cannot *enforce* that a key reaches one query. The column form couples queries through a column and can — every partition is realizable by it (Prop. 6) — and under causal decoding its exclusions are permanent (Prop. 11). Both controls may be preset or learned; the runs reported here preset them, on a softmax-one base (Miller, 2023).

What can and cannot be represented. We prove four things about this class (Sec. 3; proofs in App. G). **Separation (Thm. 1):** a scheme computing each query's row from that query's own inputs — softmax, relative-position and geometric biases, elementwise gates — cannot enforce a constraint on a *key's* degree; exclusion on the column margin is not a matter of a better score. **Conservation (Prop. 3):** no row-normalized scheme bounds every key's fan-out below n_q/n_k, and doubly stochastic attention meets that bound only by giving every key the same content-free fan-out. **The precision identity (Prop. 5):** a mechanism giving every candidate positive mass has evidence precision exactly m/n with m gold items among n, whatever the scores, so precision on an exclusive relation requires exact zeros. **Sharp recovery (Thm. 2, Cor. 2, Prop. 4):** a sparsemax program over a slice with margin δ and within-set spread W returns exactly the gold support *iff* its exclusivity lies in $[1/(W + m\delta), 1/W)$ — $\tau \geq 1/\delta$ with one gold item — sheds true members weakest-first above it, and only gains as the slice shrinks (Prop. 7); a too-large τ therefore over-sharpens. The first three bound what softmax and its query-local family can do; the fourth says what a program with exact zeros does.

Contributions. (i) A **change of represented object**, from the query–key pair to an exclusive relation carried on a margin, and on that margin by **two** predicted quantities rather than one: a **capacity** — how much influence an element commits into its relation, inherited from the base normalizer — and a **shape** — how that influence is distributed over the relation's members, *including which are exactly zero* (Sec. 2). Exclusivity is the pair: a capacity alone bounds a total without deciding membership; a shape alone redistributes a total it does not control. Of every mechanism we know of (App. C, Table 4), **MarSea is the only one whose shape cell is predicted from content on a relation whose size does not track the context**; the experiments separate that from the nearest row-margin neighbour, row sparsemax. (ii) A **theory** of the class — the four results above — which classifies existing mechanisms, MarSea's two forms among them (Sec. 3). (iii) A closed-form **mechanism** with a form per margin, selected per task, a relation anchored on the attention scores, and two controls, τ and λ, preset in the deployed configuration (Sec. 2). (iv) An **empirical demonstration** on the one-to-many case: multi-value retrieval with a 1.5B decoder, three seeds, against softmax, softmax-one, causal learned-marginal transport and row-wise sparsemax, decomposing the gain over the base normalizer by switching the relation off at inference — the inference-time part shows in

repeated and omitted values, not in exact-set — and stating plainly where the mechanism does not help (Sec. 4).

Related work. App. C treats it in full. *Geometry in scores* (Hu et al., 2018; Wu et al., 2021; Gao et al., 2021; Liu et al., 2022; Hou et al., 2024) and *differential attention* (Ye et al., 2025) are query-local, so Thm. 1 covers them. *Transport-based attention* (Sander et al., 2022; Shahbazi et al., 2025) fixes the column marginal to *uniform*, as does Slot Attention's first normalization (Locatello et al., 2020); Zhang et al. (2023) learns both marginals and is the closest prior art to MarSea's inherited capacities, but its transport is dense, so Prop. 5 pins its precision; we run it causally as a baseline. **Selective Attention (Leviathan et al., 2025) is the one mechanism outside the transport family we know of that escapes Thm. 1**; expert-choice routing (Zhou et al., 2022) makes the same move outside attention. Neither predicts from content how much a key should emit. *Sparse attention* (Martins & Astudillo, 2016; Peters et al., 2019; Correia et al., 2019) and the regularized-argmax framework (Niculae & Blondel, 2017; Blondel et al., 2020) supply our variational form, which is *not* our contribution. *Softmax-one* (Miller, 2023) is both MarSea's base normalizer and the strongest baseline (Sec. 4). Veličković et al. (2025) prove that, over a bounded input space, softmax attention must disperse as items grow; Prop. 9 makes that quantitative. *Position bias* (Liu et al., 2024) has its own accounts (Chowdhury, 2026; Wu et al., 2025b; Herasimchyk et al., 2026); we make no claim about that axis (App. D).

2 METHOD

This section demonstrates how MarSea models multivariate competitive exclusion by focusing its implementation on a common case of the general exclusive relation — one-to-many relations.

With n_q queries, n_k keys and values V_j, the scores are the layer's own,

$$s_{ij} = q_i \cdot k_j / \sqrt{d}, \tag{1}$$

MarSea replaces the normalization step of an attention layer and nothing else. It has three components — a *base normalizer*, a *relation*, and a *one-end program* that comes in a row form and a column form — and two scalar controls, τ and λ. We introduce them together so that the deployed mechanism is an instance of the classes Sec. 3 characterizes; App. A collects notation and the guarantee each component carries.

2.1 THE OBJECT: A ONE-TO-MANY RELATION, CARRIED ON ITS MANY-END

A relation $\mathcal{E} \subseteq [n_q] \times [n_k]$ is **one-to-many** when every element on one side has exactly one true partner on the other side, and an element on that other side may have many. We call the side whose elements each have one partner the **many-end** and the other the **one-end**. In list retrieval the relation is one-to-one per token and one-to-many per span, and can be read from either side: read from the query side, the answer rows are the many-end (each copies from one source span) and the source tokens the one-end — the *row form*; read from the key side, each source token is the many-end element (it is copied by exactly one answer row) and the answer rows the one-end — the *column form*. Which margin carries the program is set per task, not learned.

The program runs at the **many-end**: at each many-end element the one-end candidates compete, one wins, and the rest receive little to nothing; the one-end carries no program. The **relation is a set per many-end element** — $\mathcal{E}_{i\cdot} \subseteq [n_k]$ for a query, $\mathcal{E}_{\cdot j} \subseteq [n_q]$ for a key — and pairs outside $\mathcal{E}$ keep ordinary attention; every result of Sec. 3 is stated on such a slice. **This paper instantiates the one-to-many case, one form per task**: the case a decoder's retrieval tasks present, and one whose partition gives every metric a ground truth. Constraining both margins at once adds a consistency hypothesis on their target sets and is not evaluated here (App. D).

2.2 STEP 1 IS THE BASE NORMALIZER

Let N be a row normalizer and $A^N := N(S_{i\cdot})$ its output over the visible row scores: ordinary row-softmax, $A^N_{ij} = e^{s_{ij}} / \sum_{j'} e^{s_{ij'}}$, or softmax-one (Miller, 2023), $A^N_{ij} = e^{s_{ij}} / (1 + \sum_{j'} e^{s_{ij'}})$, under which a row may attend to nothing and $\sum_j A^N_{ij} < 1$. **The capacity is inherited, not predicted**: the

program below redistributes the mass A^N already places on a relation set. It reuses the entries A^N_{ij} and the row's log-normalizer $\mathrm{lse}_i := \log \sum_{j' \in \mathrm{vis}(i)} e^{s_{ij'}}$ (under softmax-one, $\log(1 + \sum_{j'} e^{s_{ij'}})$), so that $\log A^N_{ij} = s_{ij} - \mathrm{lse}_i$ for both.

2.3 The row form

For a many-end query i, if its relation set (the set of keys that query i admits into its relation — the top-k of the relation logit of Sec. 2.6) is empty, $\mathcal{E}_{i\cdot} = \emptyset$, the row program falls back to the base normalizer, i.e., keeps $A^N_{i\cdot}$. Otherwise, if $\mathcal{E}_{i\cdot} \neq \emptyset$, the row form is active for query i. The relation set and the non-relation set inherit their mass budgets as:

$$R_i := \textstyle\sum_{j \in \mathcal{E}_{i\cdot}} A^N_{ij}, \qquad T_i := \textstyle\sum_{j \notin \mathcal{E}_{i\cdot}} A^N_{ij}. \tag{2}$$

The program re-shapes the relation set/slice and scales the complement:

$$A_{i,\mathcal{E}_{i\cdot}} = \big(R_i + \lambda_{\mathrm{row}}\, T_i\big)\ \mathrm{sparsemax}\big(\tau_{\mathrm{row}}\, s_{i,\mathcal{E}_{i\cdot}}\big), \qquad A_{ij} = (1 - \lambda_{\mathrm{row}})\, A^N_{ij} \quad \forall\, j \notin \mathcal{E}_{i\cdot}. \tag{3}$$

Here $\mathrm{sparsemax}(z) = \arg\max_{p \in \Delta} \{\langle p, z\rangle + H^{\mathsf{T}}_2(p)\}$ with H^{T}_2 the Tsallis 2-entropy, whose solution is the truncation

$$p_{ij} \ = \ [\,\tau_{\mathrm{row}} s_{ij} - \psi_i\,]_+, \tag{4}$$

affine in the score wherever positive and *exactly* zero past the threshold ψ_i (Martins & Astudillo, 2016) — the precise sense of "little to nothing". The row form competes on the raw scores $s_{i,\mathcal{E}_{i\cdot}}$: within one row every candidate carries the same query offset, so the offset cancels inside the sparsemax (Eq. (4) is invariant to adding a constant to the row) and no normalization is needed — the point at which the column form differs (Sec. 2.4). By Prop. 1, the row total is conserved for every λ — $\lambda = 0$ leaves the complement untouched, $\lambda = 1$ moves the whole row onto the relation — and the row form is row-local, so teacher forcing and decoding coincide.

2.4 The column form

For a many-end key j with an empty relation set $\mathcal{E}_{\cdot j}$ (the queries that admit j into their relation), the column keeps A^N. Otherwise, as in the row form, the inherited masses of the relation set and its complement are

$$R_j \ := \ \sum_{i \in \mathcal{E}_{\cdot j}} A^N_{ij}, \qquad T_j \ := \ \sum_{i \notin \mathcal{E}_{\cdot j}} A^N_{ij}, \tag{5}$$

both summed over the rows generated so far that see key j (Sec. 2.7). The members share the budget $R_j + \lambda_{\mathrm{col}} T_j$ by a sparsemax, so the leader takes the column outright when its margin is clear, but **they compete on row-normalized scores**: a raw score s_{ij} carries query i's own offset, so a query with uniformly high logits would win every column it belongs to. Subtracting each row's log-normalizer removes the offset,

$$z_{ij} \ := \ \log A^N_{ij} \ = \ s_{ij} - \mathrm{lse}_i, \qquad p_{\mathcal{E}_{\cdot j}, j} \ = \ \mathrm{sparsemax}\big(\tau_{\mathrm{col}}\, z_{\mathcal{E}_{\cdot j}, j}\big), \tag{6}$$

the log of the share of its own attention that row i gives key j; z enters only through $p_{\cdot j}$, and every mass in A is read off A^N. The column form is inherently **causal**: key j's relation set grows one generated row at a time and is not bounded by k. The assignment is

$$A_{\mathcal{E}_{\cdot j}, j} \ = \ (R_j + \lambda_{\mathrm{col}} T_j)\, p_{\mathcal{E}_{\cdot j}, j}, \qquad A_{ij} \ = \ (1 - \lambda_{\mathrm{col}})\, A^N_{ij} \quad \forall i \notin \mathcal{E}_{\cdot j}. \tag{7}$$

Rows are not re-normalized afterwards: a row that wins several columns carries more than a unit. Prompt rows and keys with an empty relation set keep A^N exactly.

2.5 The two controls, and over-sharpening

$\tau > 0$ is the **shape control**: by Lem. 1 the support of $\mathrm{sparsemax}(\tau s)$ is a prefix of the sorted order, shrinking from the whole slice as $\tau \to 0^+$ to an argmax as $\tau \to \infty$. **A large τ over-sharpens**: on a slice of k candidates it is an argmax, and a source scored second receives an exact zero (Cor. 2), so relaxing τ is what the theory prescribes when the ranking inside the slice is imperfect. $\lambda \in [0, 1]$ is the **relocation control**: it moves mass between the slice and its complement, conserving the total, and is the only parameter that changes how much a many-end element commits to its relation. Both

may be preset or learned (heads $\tau_\phi^{\mathsf{K}}, \tau_\phi^{\mathsf{Q}}$ exist in the implementation); **the reported configuration presets both**: $\tau_{\text{row}} = 1$, $\lambda_{\text{row}} = 0$ for the row form, and $(\tau_{\text{col}}, \lambda_{\text{col}}) = (1, 0)$ for the column arms except one ablation at $(0.75, 0.1)$. App. I records how they were chosen, and Cor. 3 what a preset gives up.

2.6 THE RELATION

The relation decides where the program runs; three design decisions fix it, each answering a failure we met (App. I).

It is a partition, not a threshold. For a many-end query i, $\mathcal{E}_{i\cdot}$ is the top-k of a relation logit over the visible keys, $k = 8$, with a straight-through gate (App. I); every many-end row commits to at most k candidates. The column slice $\mathcal{E}_{\cdot j}$ is the same relation read along key j: the answer rows whose top-k contains j; a row's membership is fixed when the row is generated. Three scoping rules complete the relation. It exists on **answer rows only** — prompt rows keep A^N and never hold a relation, so in the column form only generated rows are members. It exists on a **subset of heads**: eight heads in three patched layers, those whose softmax attention puts at least half its mass on an answer row's own source block (App. I); every other head is plain A^N. And the **first token** — the decoder's attention sink — is excluded from every relation by construction.

It is anchored on the attention score. The logit is

$$\ell_{ij} \;=\; \mathrm{sg}(s_{ij}) \;+\; \langle Uk_j,\, Vq_i\rangle/\sqrt{r} \;+\; b_0, \qquad r = 16, \tag{8}$$

with sg the stop-gradient and U, V initialized small. The head learns the *deviation* from attention — in list retrieval, pushing already-listed sources down — rather than the ranking from scratch.

It is supervised where ground truth exists. Training sequences carry *block gold*: a partition of the prompt into blocks, each pairing a set of answer rows with the span of prompt tokens they copy from. A warm start fits the relation logits to the block gold before the program is switched on, and an auxiliary binary cross-entropy on the logits, with the other blocks' spans as hard negatives, keeps them there (App. I). **The auxiliary loss reaches only** U, V, b_0: K, Q and the anchor are detached, so it cannot change the backbone, which matters for how Sec. 4 attributes the result. Without it the language-model loss evicts the copy source from the relation within a few hundred steps.

2.7 WHAT THE FORMS GUARANTEE, AND WHAT THEY COST

Proposition 1 (What the one-end forms guarantee)**.** *For every relation $\mathcal{E}$, $\tau > 0$ and $\lambda \in [0, 1]$: **(i)** the row form conserves every row total, $\sum_j A_{ij} = \sum_j A^N_{ij}$; **(ii)** entries off the slice equal $(1-\lambda)A^N_{ij}$, so at $\lambda = 0$ no slice claim is met by moving mass out of the relation; **(iii)** $\mathcal{E} = \emptyset$ gives $A = A^N$ entrywise; **(iv)** the column form conserves every key's total over the generated rows that see it, and does not renormalize rows; **(v)** the row form is query-local (Def. 1) and the column form is not; **(vi)** members past the sparsemax threshold receive exact zeros, so support precision and recall (Def. 2) are defined on the slice.*

Part (v) places the row form inside the query-local class of Sec. 3 and the column form outside it; (vi) places both outside the full-support class of Prop. 5; (iii) makes the relation-off control of Sec. 4 exact. **Cost and causal decoding.** A row program sorts at most k entries, $O(k \log k)$ per many-end element, and is row-local, so teacher forcing and decoding coincide. A column's slice grows with the generated rows and is not bounded by k; at fixed τ its threshold is non-decreasing, so *exclusion is permanent* while admission is provisional, and the frozen-prefix decode path over-admits and never under-admits (Prop. 11). Implementation and cost: App. B.

3 THEORY OF COMPETITIVE EXCLUSION IN ATTENTION

Every claim in this section is proved in App. G, which restates each result before proving it; the few given here in prose are stated formally there. The results concern classes of attention schemes, not one mechanism: Sec. 3.1 characterizes what query-local and row-normalized schemes can enforce on a key's degree, and Secs. 3.2–3.4 characterize exact support recovery by a sparsemax over an index set and the precision of every full-support scheme. A result stated on a many-end slice holds on either margin with the roles of i and j exchanged; MarSea's two forms are its instances.

3.1 TWO OBSTRUCTIONS: QUERY-LOCAL AND ROW-NORMALIZED ATTENTION

Definition 1 (Query-locality, and product form)**.** *Write query i's inputs as $x_i := (q_i, \gamma_{i\cdot}) \in \mathbb{R}^d \times \Gamma^{n_k}$ with admissible set $\mathcal{D}_i$. A scheme is **query-local** if for every i the row $A_{i\cdot}$ is a function of $(x_i, \{k_j\}_j)$ alone, independent of $x_{i'}$ for $i' \neq i$. A constraint $\mathcal{C} \subseteq \mathbb{R}^{n_q}_{\geq 0}$ is **product-form** if $\mathcal{C} = \prod_i \mathcal{C}_i$; a capacity bound $\sum_i x_i \leq c$ is not, since whether one coordinate is admissible depends on the others. The first is independence in the* mechanism*, the second in the* constraint*.*

Row-softmax, softmax-one, every additive relative-position or geometric bias, every elementwise gate $A_{ij} = \alpha_{ij} g(q_i, k_j, \gamma_{ij})$, row-entmax — and **MarSea's row form** — are query-local. MarSea's column form is not.

Theorem 1 (Query-local schemes attain exactly a product set)**.** *Fix a key j, evaluate a query-local scheme over $\mathcal{D} = \prod_i \mathcal{D}_i$ with $\{k_j\}$ fixed, and put $\mathcal{R}_i := \{A_{ij}(x_i) : x_i \in \mathcal{D}_i\}$. Then the attainable set of fan-out columns is* exactly *the product,*

$$\big\{A_{\cdot j}(x) : x \in \mathcal{D}\big\} \;=\; \prod_{i=1}^{n_q} \mathcal{R}_i, \tag{9}$$

and the scheme satisfies $A_{\cdot j} \in \mathcal{C}$ for all $x \in \mathcal{D}$ iff $\prod_i \mathcal{R}_i \subseteq \mathcal{C}$. Hence no query-local scheme enforces a non-product-form $\mathcal{C}$ except vacuously.

Assumption 1 (Non-degenerate replication)**.** *Fix key j. There exist $a > 0$ and $M \leq n_q$ such that some admissible input has at least M queries i with $A_{ij}(x_i) \geq a$. We call (a, M) a* replication pair *for key j.*

Corollary 1 (Cardinality fails, and fails on real inputs)**.** *Cardinality $\mathcal{C}_{\text{card}} = \{x : |\operatorname{supp}(x)| \leq m\}$ with $1 \leq m < |\mathcal{E}_{\cdot j}|$ and capacity $\mathcal{C}_{\text{cap}} = \{x : \sum_i x_i \leq c\}$ with $c > 0$ are both non-product-form for $|\mathcal{E}_{\cdot j}| \geq 2$, so by Thm. 1 no query-local scheme enforces either. The failure is realized on constrained input spaces: given a replication pair (a, M) inside the set (Assumption 1), some admissible input has M coordinates at least a, so **cardinality fails whenever** $m < M$ and capacity whenever $c < Ma$.*

Which form is which. Each key reaching one query is a cardinality constraint on a column ($m = 1$), non-product-form by Cor. 1. The row form is query-local (Prop. 1(v)), so by Thm. 1 it can *learn* a partition and *select* within a row but cannot enforce the constraint; the column form decides a column's entries jointly and enforces it by construction. Prop. 2 states the split sharply.

Proposition 2 (Selection is local; scope-respecting selection is not)**.** *Fix a query-local scheme. **(i)** Selection at a* single *target is available to it: row-sparsemax at large temperature admits only $\arg\max_j s_{ij}$ at every target. **(ii)** Scope is not: for a scope bound κ_j (the requirement being $|T_j| \leq \kappa_j$), a replication pair (a, M) with $\kappa_j < M$ drives $|\operatorname{supp}(A_{\cdot j})| \geq M > \kappa_j$ on some admissible input. **(iii)** Hence the non-local content is neither in isolation but their* conjunction*, which the column form satisfies by construction and the row form does not.*

The hypothesis is generic: near-equivalent candidates — distractors retrieved for the same question, earlier mentions of an already-listed value — are what a query-local rule must treat alike. So a gain of the row form over softmax is a gain in *selection within a row*, not in coupling, and what separates the column form from every row-local rule is how many owners a key's column reaches (Sec. 4).

Proposition 3 (Conservation obstruction)**.** *If $\sum_j A_{ij} = 1$ for every query, then $\max_j \sum_i A_{ij} \geq n_q/n_k$: no row-normalized scheme can bound* every *key's fan-out below the query-to-key ratio, and doubly stochastic attention attains it with equality for every key.*

Neither obstruction implies the other: Prop. 3 forbids bounding all columns by a common constant but not a single one, while Thm. 1 forbids a query-local scheme from bounding even one; together they require query coupling *and* a learned, non-uniform column marginal. Prop. 3 bites where a small key set serves many queries and is weak in self-attention ($n_q = n_k$); it rules out the transport remedy, while Thm. 1 carries the argument in the decoder of Sec. 4. It applies to MarSea's inherited totals too: MarSea decides which members of a relation the total reaches, not the total.

3.2 THE PREFIX STRUCTURE OF SPARSEMAX

Sparsemax has one structural property everything else rests on. With $G(k) := \sum_{r\le k}(z_{(r)} - z_{(k)})$ non-decreasing in k, the support of $\mathrm{sparsemax}(z)$ is exactly the $k^\star$ largest entries with $k^\star = \max\{k : G(k) < 1\}$: a *prefix* of the sorted order (Lem. 1). It gives *monotone exclusivity* — $k^\star(\tau)$ is non-increasing, from the whole slice at $\tau \to 0^+$ to $|\arg\max s|$ at $\tau \to \infty$ (the shape control of Sec. 2.5) — permanence of exclusion as a prefix grows (Prop. 11), and the endpoints of Thm. 2, which G computes.

3.3 EXACT SUPPORT RECOVERY, AND WHAT A PRESET EXCLUSIVITY GIVES UP

Write the slice as a vector of scores s of length $|\mathcal{E}|$ with a target set T of size m — the true partner(s) of the many-end element — so that on a row T is the source key(s) and on a column the owner row(s).

Assumption 2 (Margin and spread)**.** $\delta := \min_{i\in T} s_i - \max_{i\notin T} s_i > 0$*; and* $W := \sum_{i\in T}(s_i - \min_{i'\in T} s_{i'})$*, which is* ≥ 0 *by definition rather than by hypothesis.*

Theorem 2 (Exact support recovery, with sharp endpoints)**.** *Let Assumption 2 hold on a slice. Then* $\mathrm{supp}(\mathrm{sparsemax}(\tau s)) = T$ if and only if

$$\tau \in \left[\ \tfrac{1}{W+m\delta},\ \tfrac{1}{W}\ \right) \qquad (1/W := \infty \text{ if } W = 0), \tag{10}$$

and this interval is non-empty for every *slice with a positive margin, since* $m\delta > 0$*.*

Both endpoints are exact because $G(m{+}1) = \tau(W{+}m\delta)$ is computed, not bounded. Exact recovery is never obstructed on a slice; the obstruction is *width*, $m\delta/W$ relative.

Corollary 2 (One true partner, and a preset τ)**.** *If* $m = 1$ *then* $W = 0$ *and the interval is* $[1/\delta, \infty)$*: a preset exclusivity* τ *recovers the partner exactly iff* $\delta \ge 1/\tau$*, and for every such* τ *the program puts the* entire *inherited mass on the partner and an exact zero on every other member. If instead the partner is mis-ranked,* $\delta \le 0$*, then no* τ *recovers it; as* $\tau \to \infty$ *the program puts the whole mass on the wrong member and an exact zero on the partner, while as* $\tau \to 0^+$ *the partner keeps a* $1/|\mathcal{E}|$ *share.*

In particular, the over-sharpening of Sec. 2.5 is this corollary at large τ: on a slice of $k = 8$ the program becomes an argmax over eight and a source scored second is lost entirely, whereas at $\tau = 1$ a row with $\delta \ge 1$ is still recovered and a mis-ranked row keeps a share.

Corollary 3 (What a preset exclusivity gives up)**.** *If two slices satisfy* $W_2 + m_2\delta_2 \le W_1$ *their recovery intervals are disjoint, so no single* τ *achieves exact recovery on both; a content-predicted* τ *can. With* $m = 1$ *throughout the intervals are nested,* $[1/\delta, \infty)$*, and one preset* τ *recovers every slice with* $\delta \ge 1/\tau$ *at once.*

With one true partner there is no within-target spread to trade against the margin, which is why a preset is tenable in the one-to-many case; the slices it fails, $\delta < 1/\tau$, a learned τ reading the slice $(\tau_\phi^{\mathsf{K}}, \tau_\phi^{\mathsf{Q}})$ would recover, at the price of a parameter to find.

Proposition 4 (Graded degradation, weakest-first)**.** *Past the upper endpoint of its interval a slice sheds true members in* non-decreasing *order of strength — one at a time except that tied members shed together — with precision pinned at* 1 *throughout: for* $\tau > 1/W$*,* $P = 1$ *and* $R = k(\tau)/m$ *with* $k(\tau) = \max\{k : W(k) < 1/\tau\}$*,* $W(k) := \sum_{r\le k}(s_{(r)} - s_{(k)})$*.*

3.4 FIDELITY ON THE MANY-END: THE PRECISION IDENTITY

Reading the allocation as retrieval of a ground-truth relation separates the two error types, which is possible only once exact zeros exist: under a normalized full-support allocation, mass-precision and mass-recall coincide.

Definition 2 (Support precision and recall on a slice)**.** *For a many-end element with slice* $\mathcal{E}$*, target set* $T \subseteq \mathcal{E}$*,* $m = |T|$ *and realized support* $S := \mathrm{supp}(A_{\mathcal{E}})$ *on the relation, put* $P = |S \cap T|/|S|$*,* $R = |S \cap T|/m$ *and* $\rho = \sum_{i\in T} p_i$*, with* $P = 1$ *on an empty support. Targets outside* $\mathcal{E}$ *are*

relation-recall *misses, $|T \cap \mathcal{E}|/m$, reported separately so that a relation error is never charged to τ.*

Proposition 5 (The precision ceiling of full support)**.** *Let a mechanism assign $p_i > 0$ to every member of a slice — row-softmax, softmax-one, any additive bias on either, any elementwise gate with positive factors, any entropy-regularized transport. Then for every slice, every score configuration and every temperature, $P = m/|\mathcal{E}|$* exactly. *It is independent of the scoring function, so it improves by no training, no architecture within the class and no temperature; on the unrestricted margin it is m/n and tends to 0 as the candidate set grows, while recall is 1 trivially.*

A full-support mechanism can rank the targets perfectly without moving its precision, so precision on an exclusive relation is controlled only by exact zeros. For a sparsemax slice the recovery interval is exactly a precision–recall interval (Prop. 8), while a softmax slice needs τ growing like $\log|\mathcal{E}|$ to hold the off-target mass at a fixed fraction (Prop. 9); App. E gives the full account.

4 Experiments

Every arm is the same decoder, patched layers, adapters, data and schedule with a different normalizer — a mechanism ablation, not a leaderboard entry.

Setup. Qwen2.5-1.5B (Apache 2.0) with LoRA adapters on every layer; the normalizer is replaced in the four layers with the backbone's top retrieval heads (Wu et al., 2025a), $\{14, 19, 23, 24\}$, and the relation exists on eight of their heads (Sec. 2.6). Every arm and seed starts from the same 500-step softmax warm start and trains $2{,}000$ further steps at $L = 4{,}096$ on one mixture: RULER (Hsieh et al., 2024) multi-value and multi-query×multi-value retrieval and variable tracking, with MuSiQue (Trivedi et al., 2022) and HotpotQA (Yang et al., 2018) questions. We evaluate on ten held-out 4K RULER retrieval sets, greedy, grouped by the number of values m a correct answer lists; at $m \leq 8$ every arm and seed scores 1.00, so Table 1 reports $m = 16$ (75 lists per seed) and $m = 32$ (50). Per list we score *exact-set* (returned set = gold set), per-value precision and recall, and the *repeated* and *omitted* values, since exact-set does not see repeats and exclusion acts on particular kinds. Three seeds per arm; re-generating a checkpoint moves its exact-set by 0.02–0.04 (greedy bf16 decoding is not bit-reproducible), so differences below about 0.05 are not separable and none is called significant (App. I).

Arms. Baselines: **B0**, row softmax; **B1**, softmax-one (Miller, 2023); **B2**, a learned-marginal transport in the form of Zhang et al. (2023), made causal; **B3**, row sparsemax (Martins & Astudillo, 2016), i.e. α-entmax at $\alpha = 2$. **MarSea** is the row form on the softmax-one base, $\tau_{\text{row}} = 1$, $\lambda_{\text{row}} = 0$, $k = 8$; the same on the softmax base is reported beside it. The **column form**, the ablation of which margin carries the program, runs on the same lists with the value tokens as the many-end, competing on raw scores s (softmax base, two seeds; softmax-one base) or on the row-normalized z (softmax-one base) at $(\tau_{\text{col}}, \lambda_{\text{col}}) = (1, 0)$ (a softer-preset ablation is in App. I). Every MarSea arm is also evaluated with **the relation off** ($\mathcal{E} = \emptyset$, exactly its base normalizer by Prop. 1(iii)): on minus off is the program's inference-time effect; off minus the same-base baseline is the training-time one.

MarSea against the baselines. MarSea has the highest three-seed exact-set at both list lengths, the highest recall (tied with B1 at 16 values), and the fewest repeated and omitted values of any baseline at 32 values (0.21 vs B1's 0.31; 0.52 vs 0.56). It raises exact-set over softmax by 0.062 and 0.133. Over softmax-one its exact-set margins ($+0.009$, $+0.020$) are inside seed variance, and **we do not claim it outperforms softmax-one**: what separates the two is which errors remain, not how many lists are right.

What the program does at inference, and what training through it does. Measured against its own base normalizer, the gain splits differently on the two bases. *On softmax-one*, the weights with the relation off score what B1 scores (0.929 / 0.727), with fewer repeats at 32 values than B1 (0.20 vs 0.31) but more omissions (0.76 vs 0.56); the program at inference then takes omissions to 0.52 and recall from .976 to .984 and leaves repeats where they were. *On softmax*, the weights carry the exact-set gain over B0 (0.920 / 0.687 with the relation off against 0.871 / 0.627), and the program at inference removes a quarter of the remaining repeats ($0.23 \rightarrow 0.17$) and of the omissions

Table 1: **Multi-value retrieval at 4K, step 2,500.** Exact-set (seed mean ± sd), precision / recall, repeated and omitted values per list; n seeds; "off": relation off. **Bold**: best three-seed arm, off rows excluded. Per seed: Table 10.

		$m = 16$				$m = 32$			
arm	n	exact	prec / rec	rep	omit	exact	prec / rec	rep	omit
B0 softmax	3	$0.871 \pm .031$	.991 / .992	0.09	0.13	$0.627 \pm .025$	.982 / .967	0.43	1.05
B1 softmax-one	3	$0.924 \pm .006$	**.995 / .995**	**0.03**	**0.08**	$0.740 \pm .033$	.987 / .982	0.31	0.56
B2 causal MESH	3	$0.827 \pm .050$	.984 / .980	**0.03**	0.32	$0.620 \pm .049$	.974 / .964	0.40	1.15
B3 row sparsemax	3	$0.787 \pm .000$	.986 / .986	0.13	0.22	$0.387 \pm .075$	.966 / .957	0.78	1.38
Row form ($\tau_{\text{row}} = 1$, $\lambda_{\text{row}} = 0$)									
MarSea (softmax-one base)	3	$\mathbf{0.933} \pm .011$	**.995 / .995**	0.04	**0.08**	$\mathbf{0.760} \pm .043$	.990 / **.984**	0.21	**0.52**
off (= B1 at inference)	3	$0.929 \pm .043$	.994 / .994	0.06	0.08	$0.727 \pm .038$	.989 / .976	0.20	0.76
softmax base	3	$0.916 \pm .072$	**.995** / .982	**0.03**	0.29	$0.707 \pm .062$	**.992** / .969	**0.17**	1.00
off (= B0 at inference)	3	$0.920 \pm .019$	.994 / .983	0.04	0.27	$0.687 \pm .077$	.990 / .959	0.23	1.33
Column form (base, competing score; τ_{col}, λ_{col})									
softmax, s; 1, 0	2	$0.860 \pm .010$	.987 / .988	0.07	0.15	$0.800 \pm .020$	.993 / .981	0.12	0.28
off	2	$0.920 \pm .010$	.994 / .994	0.07	0.10	$0.770 \pm .070$	.993 / .983	0.20	0.42
softmax-one, s; 1, 0	3	$0.907 \pm .012$	.991 / .992	0.08	0.13	$0.640 \pm .016$	.986 / .973	0.29	0.85
off	3	$0.924 \pm .005$	.991 / .992	0.12	0.13	$0.627 \pm .009$	.977 / .965	0.68	1.13
softmax-one, z; 1, 0	3	$0.924 \pm .025$	**.995 / .995**	0.04	**0.08**	$0.720 \pm .043$	.990 / .980	0.22	0.63
off	3	$0.956 \pm .013$	.998 / .995	0.02	0.08	$0.713 \pm .025$	.989 / .981	0.23	0.61

($1.33 \to 1.00$) at 32 values. On both bases the program's inference-time effect is on the 32-value errors, within noise on exact-set, and the weight effect can only come from the language-model loss: the auxiliary block loss reaches U, V, b_0 alone. Softmax-one withholds mass from every key, exclusion from particular keys; combined, they keep softmax-one's omissions and lower its repeats.

Which margin carries the program. The column form enforces what the row form can only learn: a value token committed to the row that copied it is excluded from later rows (Prop. 11). On raw scores it does so on every seed — on softmax-one, 32-value repeats fall from 0.68 to 0.29 and omissions from 1.13 to 0.85 — yet exact-set stays at softmax's level (0.640), because the weights trained through it are no better than B0's; row-trained weights run through the column program do worse still (0.890 / 0.607 vs 0.929 / 0.727 off). A probe (App. I) locates the failure: the column relations hold the copy source nearly as often as the row relations, but only 0.59–0.63 of a value's mass lands on its copying row, against 0.72–0.74 for the row program. On raw scores each member carries its row's offset, so the margin Thm. 2 asks of a slice is partly a margin in offsets; the row-normalized z removes them (Sec. 2.4). **Changing only the score** (z for s; softmax-one, $(1, 0)$) lifts the landed mass to 0.70–0.71, level with the row program, and repairs the weights trained through the column form: off, they are the study's best at 16 values (0.956) and at 32 rise from 0.627 to 0.713, beside the row form's (0.727) and B1 (0.740). The program at inference then adds nothing: on minus off is -0.032 at 16 values, on every seed, and $+0.007$ at 32 (seed signs $+, -, +$), with repeats unchanged (0.22 vs 0.23) — once the copying row wins its column, what the program withholds from later rows they were no longer copying. At 0.924 / 0.720 the column form stays below the row form, so on this task its effect is through training, not inference, and the row form is the deployed MarSea. This is the single-layer caveat of App. D: in a decoder the query already carries what has been listed, so a row-local rule can avoid re-listing and enforcement has little left to remove. On softmax with raw scores the column form has the study's best 32-value exact-set (0.800) and fewest repeats (0.12), on two seeds from which we draw nothing.

Other arms, and what is not evidence. Row sparsemax (B3), exact zeros over a whole row with no relation to scope them, is the weakest arm on exact-set; causal MESH (B2), the dense learned-marginal mechanism (App. C), is at or below softmax and produces the most extraneous values (0.23 / 0.44 per list): sparsity and learned marginals are not the operative ingredients. Variable tracking is saturated (≥ 0.96 exact-set at 4K and the untrained 8K, no repeats; Table 9). Multi-hop QA is in the training mixture but not an evaluation: a $k = 8$ relation reaches about 2% of a supporting span's tokens (App. I). The presets come from single-seed pilots that are not evidence. Three seeds, one backbone, one training length, LoRA, and relation heads chosen from the trained softmax arm bound all of the above (App. D).

Reproducibility statement

All analytical claims are stated with explicit hypotheses and proved in App. G; each is additionally checked by a self-contained numpy program supplied as supplementary material, which seeds each block separately, runs in under a minute on one CPU core, and reproduces every figure in Tables 8 and 7. **Every claim verifies at zero violations**, with the count reported *per claim* rather than as one headline number, because the rows are not commensurable and not all of them could have failed: a dagger marks every row whose conclusion follows from how the generator builds its input, and App. H says what each does and does not test. The harness checks the theorems as stated — sparsemax on an index set — and predates the deployed configuration; it is not evidence about real data. Code for MarSea's two forms, the baselines, data generation, training, evaluation, the relation probe and the harness is provided as anonymized supplementary material, with the configuration file of every reported arm and a README that maps the harness output to the paper's numbering. Every experimental number in Sec. 4 is reproducible from that code, the configuration files and the evaluators listed in App. I, which also gives the per-seed values behind every mean; the training pool is regenerated from named generator seeds. All datasets are public.

AI use statement

Large language model assistance was used for literature search, LaTeX editing, consistency checking, and code review of the experimental pipeline. All theorem statements, proofs, and experimental design decisions are the authors'; every claim retained was verified by the authors, and the numerical verification and the experiments were executed and their output inspected rather than reported from a model's description of it.

References

Quentin Berthet, Mathieu Blondel, Olivier Teboul, Marco Cuturi, Jean-Philippe Vert, and Francis Bach. Learning with differentiable perturbed optimizers. In *Advances in Neural Information Processing Systems*, 2020.

Mathieu Blondel, André F. T. Martins, and Vlad Niculae. Learning with Fenchel–Young losses. *Journal of Machine Learning Research*, 21(35):1–69, 2020.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, 2020.

Borun D. Chowdhury. Lost in the middle at birth: An exact theory of transformer position bias. *arXiv preprint arXiv:2603.10123*, 2026.

Gonçalo M. Correia, Vlad Niculae, and André F. T. Martins. Adaptively sparse transformers. In *Conference on Empirical Methods in Natural Language Processing*, 2019.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 2022.

Peng Gao, Minghang Zheng, Xiaogang Wang, Jifeng Dai, and Hongsheng Li. Fast convergence of DETR with spatially modulated co-attention. In *IEEE International Conference on Computer Vision*, 2021.

Hanna Herasimchyk, Robin Labryga, Tomislav Prusina, and Sören Laue. A structural theory of position bias in transformers. *arXiv preprint arXiv:2602.16837*, 2026.

Xiuquan Hou, Meiqin Liu, Senlin Zhang, Ping Wei, Badong Chen, and Xuguang Lan. Relation DETR: Exploring explicit position relation prior for object detection. In *European Conference on Computer Vision*, 2024.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. RULER: What's the real context size of your long-context language models? In *First Conference on Language Modeling (COLM)*, 2024.

Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei. Relation networks for object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. GShard: Scaling giant models with conditional computation and automatic sharding. In *International Conference on Learning Representations*, 2021.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Selective attention improves transformer. In *International Conference on Learning Representations*, 2025.

Shilong Liu, Feng Li, Hao Zhang, Xiao Yang, Xianbiao Qi, Hang Su, Jun Zhu, and Lei Zhang. DAB-DETR: Dynamic anchor boxes are better queries for DETR. In *International Conference on Learning Representations*, 2022.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173, 2024.

Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. In *Advances in Neural Information Processing Systems*, 2020.

André F. T. Martins and Ramón F. Astudillo. From softmax to sparsemax: A sparse model of attention and multi-label classification. In *International Conference on Machine Learning*, 2016.

Evan Miller. Attention is off by one. `https://www.evanmiller.org/attention-is-off-by-one.html`, 2023.

Vlad Niculae and Mathieu Blondel. A regularized framework for sparse and structured neural attention. In *Advances in Neural Information Processing Systems*, 2017.

Ben Peters, Vlad Niculae, and André F. T. Martins. Sparse sequence-to-sequence models. In *Annual Meeting of the Association for Computational Linguistics*, 2019.

Sara Sabour, Nicholas Frosst, and Geoffrey E. Hinton. Dynamic routing between capsules. In *Advances in Neural Information Processing Systems*, 2017.

Michael E. Sander, Pierre Ablin, Mathieu Blondel, and Gabriel Peyré. Sinkformers: Transformers with doubly stochastic attention. In *International Conference on Artificial Intelligence and Statistics*, 2022.

Clayton Sanford, Daniel Hsu, and Matus Telgarsky. Representational strengths and limitations of transformers. In *Advances in Neural Information Processing Systems*, 2023.

Ashkan Shahbazi, Chayne Thrash, Yikun Bai, Keaton Hamm, Navid NaderiAlizadeh, and Soheil Kolouri. LOTFormer: Doubly-stochastic linear attention via low-rank optimal transport. *arXiv preprint arXiv:2509.23436*, 2025.

Yi Tay, Dara Bahri, Liu Yang, Donald Metzler, and Da-Cheng Juan. Sparse sinkhorn attention. In *International Conference on Machine Learning*, 2020.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. MuSiQue: Multi-hop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022.

Petar Veličković, Christos Perivolaropoulos, Federico Barbero, and Razvan Pascanu. Softmax is not enough (for sharp size generalisation). In *International Conference on Machine Learning*, 2025.

Kan Wu, Houwen Peng, Minghao Chen, Jianlong Fu, and Hongyang Chao. Rethinking and improving relative position encoding for vision transformer. In *IEEE International Conference on Computer Vision*, 2021.

Wenhao Wu, Yizhong Wang, Guangxuan Xiao, Hao Peng, and Yao Fu. Retrieval head mechanistically explains long-context factuality. In *International Conference on Learning Representations*, 2025a.

Xinyi Wu, Yifei Wang, Stefanie Jegelka, and Ali Jadbabaie. On the emergence of position bias in transformers. In *International Conference on Machine Learning*, 2025b.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *International Conference on Learning Representations*, 2024.

Yujia Xie, Hanjun Dai, Minshuo Chen, Bo Dai, Tuo Zhao, Hongyuan Zha, Wei Wei, and Tomas Pfister. Differentiable top-k with optimal transport. In *Advances in Neural Information Processing Systems*, 2020.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380, 2018.

Tianzhu Ye, Li Dong, Yuqing Xia, Yutao Sun, Yi Zhu, Gao Huang, and Furu Wei. Differential transformer. In *International Conference on Learning Representations*, 2025.

Tongtian Yue, Longteng Guo, Jie Cheng, Xuange Gao, Hua Huang, and Jing Liu. Ada-K routing: Boosting the efficiency of MoE-based LLMs. In *International Conference on Learning Representations*, 2025.

Yan Zhang, David W. Zhang, Simon Lacoste-Julien, Gertjan J. Burghouts, and Cees G. M. Snoek. Unlocking slot attention by changing optimal transport costs. In *International Conference on Machine Learning*, 2023.

Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Zhao, Andrew Dai, Zhifeng Chen, Quoc Le, and James Laudon. Mixture-of-experts with expert choice routing. In *Advances in Neural Information Processing Systems*, 2022.

A Notation and component summary

Table 2 lists the components against the guarantee each carries; Table 3 collects every symbol used in the paper.

Table 2: The components. Every guarantee is scoped to a many-end slice of the relation.

	acts on	**parameters**	**guarantee**
Base normalizer N (§2.2)	every row	none (softmax) or none (softmax-one)	step 1; $\mathcal{E} = \emptyset \Rightarrow A = A^N$
Row form (§2.3)	$A_{i,\mathcal{E}_{i\cdot}}$ for many-end query i	τ, λ (preset or τ^{Q}_{ϕ})	Prop. 1(i)(ii)(v); Thm. 2; query-local
Column form (§2.4)	$A_{\mathcal{E}_{\cdot j},j}$ for many-end key j	τ, λ (preset or τ^{K}_{ϕ})	Prop. 1(iv)(v); Thm. 2; Prop. 11; query-coupled
Relation (§2.6)	which pairs enter	$U, V \in \mathbb{R}^{r\times d}$, b_0; k	top-k per many-end element; Prop. 7

B Cost, the causal form of the column program, and hierarchical selection

Row form. Each program sorts a slice of at most k entries, so it costs $O(k \log k)$ per many-end element, and the relation logits cost one rank-r inner product alongside the scores. The top-k bounds the program size directly and does not grow with the context. Because the row form is row-local, it is exact under any decoding and teacher forcing and decoding coincide.

Table 3: Notation. Read each row as a left pair, then a right pair.

Symbol	Meaning	Symbol	Meaning
Inputs and scores			
$n_q,\ n_k$	numbers of queries, keys	γ_{ij}	relational descriptor for (i, j)
$q_i, k_j \in \mathbb{R}^d$	query and key vectors	$b(\gamma_{ij})$	relational bias in the score
V_j	value vector of key j	$\mathcal{A},\ b$	competing set; what it bears on
s_{ij}	score, Eq. (1)	O_i	$= \sum_j A_{ij} V_j$; output for query i
The relation and the base, Secs. 2.1–2.2			
$\mathcal{E} \subseteq [n_q] \times [n_k]$	the relation; $\mathcal{E}_{i\cdot}, \mathcal{E}_{\cdot j}$ its row and column slices	many-end, one-end	the side whose elements each have one true partner; the other
N, A^N	base normalizer and its output (softmax or softmax-one)	ℓ_{ij}	relation logit, Eq. (8); $\mathcal{E}_{i\cdot}$ = its top-k
k	relation size per many-end element (8)	U, V, b_0; r	relation head and its rank (16)
The one-end programs, Secs. 2.3–2.5			
$R_i,\ T_i$	base mass on and off $\mathcal{E}_{i\cdot}$, Eq. (2)	$R_j,\ T_j$	the same for a key over $Q_j(t)$, Eq. (5)
$\tau > 0$	*exclusivity*, the shape control	$\lambda \in [0, 1]$	the relocation control
Δ	the probability simplex	H_2^{T}	Tsallis 2-entropy; sparsemax's regularizer
p, ψ	sparsemax allocation and threshold, Eq. (4)	$Q_j(t)$	answer rows generated so far that see key j
$\tau_\phi^{\mathsf{K}}, \tau_\phi^{\mathsf{Q}}$	the optional learned exclusivity heads	sg	stop-gradient
Expressivity separation, Sec. 3			
x_i	$= (q_i, \gamma_{i\cdot})$, Def. 1	$\mathcal{R}_i$	values of A_{ij} as x_i varies, at fixed j
$\mathcal{D}_i,\ \mathcal{D}$	admissible inputs; $\mathcal{D} = \prod_i \mathcal{D}_i$	(a, M)	replication pair, Assumption 1
$\mathcal{C}$	constraint set on a column	κ_j	scope bound asked of key j
Sorted order and recovery, Sec. 3.3			
$z_{(r)}$	r-th largest entry of z	$G(k),\ k^\star$	prefix statistic; support size, Lem. 1
$T,\ m$	target set of a slice and its size	δ	*margin* $\min_T s - \max_{\notin T} s$
W	*within-set spread* $\sum_T (s - \min_T s) \geq 0$; zero iff the targets tie, automatic at $m = 1$	$k(\tau)$	surviving true members above the interval, Prop. 4
Fidelity, Def. 2			
S	realized support on the slice	$P,\ R,\ \rho$	support precision, recall, target mass
Experiments, Sec. 4			
block gold	the partition of a prompt into (answer rows, span) blocks	exact-set	returned value set equals the gold set
relation off	the same weights with $\mathcal{E} = \emptyset$ at inference	repeated / omitted / extra	the error anatomy per list

Column form. The slice of key j is the set of answer rows that admit j, which k does not bound. The implementation keeps each column's $K_{\text{ret}} = 64$ best-scoring members and re-solves the column's sparsemax when a new member arrives; a training step of the column form takes about $1.5\times$ that of the row form on our runs (17–20 s against 14 s on one H200). The retention cap is harmless whenever the realized support size $k^\star$ is at most 64: by Lem. 1 the support is a prefix of the sorted order, so a member that falls out of the best 64 already had an exact zero.

The causal form. The column program runs over a set $Q_j(t)$ that grows as rows are generated, and Prop. 11 records how: at fixed τ and with membership sealed on arrival, the threshold $\psi_j^{(t)}$ is non-decreasing, so *exclusion is permanent* while admission is provisional, and the incremental *frozen-prefix* variant — which keeps each row's entry as computed when it arrived — over-admits and never under-admits. A streaming implementation therefore need only ever *evict*, and the frozen-prefix error is on the side a later row can still remove. The deployed decode path is the frozen prefix. In a decoder the column form's cost is per generated row, since $Q_j(t)$ grows; the row form's is constant.

Hierarchical selection is not needed. The hierarchical top-K construction of an earlier version — retaining the top K per block and solving one exact program at the root, exact whenever $K \geq k^\star$ (Prop. 10) — is not used. The proposition stands as a result about sparsemax, and it also shows why a fixed relation size is safe: the sparsemax support is a *prefix* of the sorted order (Lem. 1), so a relation that admits the top-k by score loses nothing the program would have kept whenever $k \geq k^\star$, and the realized $k^\star$ on exclusive slices is small (App. H). What k costs is coverage where the true partner set is large: a supporting span of hundreds of tokens against $k = 8$ is why multi-hop QA is not an evaluation here (Sec. 4).

C Related work in full

The modelling object, and where every mechanism sits. Contribution (i) describes an exclusive relation on a margin by two predicted quantities: a **capacity**, how much influence an element commits into its relation, and a **shape**, how that influence is distributed over the relation's members, including which are exactly zero. Table 4 places every mechanism we know of on these cells for both margins; the paragraphs below take the rows in turn.

Table 4: **The modelling object, and where every mechanism we know of sits in it.** A relation is carried by a margin, and on a margin it takes *two* predicted quantities: a **capacity** — how much influence the node commits or absorbs — and a **shape** — how that influence is distributed over its incident edges, including which are *exactly* zero. Capacity without shape bounds a total but does not decide membership; shape without capacity redistributes a total the mechanism does not control. "learned(k)" means a capacity predicted from the participant alone; "inherited" means taken from what the base normalizer allocates, which is what makes an exclusive margin commensurate with a non-exclusive one (Prop. 1). MarSea's **row form** fills the fan-in shape cell and its **column form** the fan-out shape cell; each inherits its capacity. Row-entmax (B3 runs it at $\alpha = 2$, row sparsemax) is the nearest neighbour on the fan-in margin — it also produces exact zeros — and differs in that its support ranges over the whole row rather than a relation of fixed size, with no conservation of the complement; Sec. 4 separates the two. Only the shape cells produce the exact zeros that Prop. 5 keys on, which is why every "dense" row is pinned at m/n however well its capacities are learned. † marks a mechanism that runs on a declared *part* of a margin rather than all of it.

	fan-out margin (per key)		**fan-in margin (per query)**	
Mechanism	capacity	shape	capacity	shape
Row-softmax	free	dense	$= 1$ fixed	dense
Geometric bias / pairwise gate	free	dense	≤ 1 fixed	dense
$\mathrm{softmax}_1$, learnable sink	free	dense	≤ 1 fixed const.	dense
Differential attention	free	dense, signed	$= 1 - \lambda$ fixed	dense, signed
Row-entmax, top-k row	free	dense	free	zeros, α fixed, whole row
Adaptively sparse	free	dense	free	zeros, α per head, whole row
Selective Attention	free	dense	$= 1$ fixed	dense
Sinkformer, doubly stochastic	$= n_q/n_k$ uniform	dense	$= 1$ fixed	dense
LOTFormer	= uniform	dense	$= 1$ fixed	dense
Slot Attention	$= 1$ uniform	dense	$= 1$ fixed	dense
MESH	= learned(k)	dense, global λ	= learned(q)	dense, global λ
Expert-choice routing	= fixed k, content-free	zeros, hard top-k	—	—
MarSea, row form†	—	—	inherited, on $\mathcal{E}_{i\cdot}$	**zeros**, τ on $\lvert\mathcal{E}_{i\cdot}\rvert \leq k$
MarSea, column form†	inherited, on $\mathcal{E}_{\cdot j}$	**zeros**, τ on $\lvert\mathcal{E}_{\cdot j}\rvert \leq k$	—	—

Geometry in scores, and other query-local remedies. Relation modules over box geometry, directional and contextual relative-position encodings, content-predicted spatial modulation, anchor-modulated positional attention and explicit position-relation priors (Hu et al., 2018; Wu et al., 2021; Gao et al., 2021; Liu et al., 2022; Hou et al., 2024) all inject relational information into the *score* (iRPE also into the value aggregation). Every one is query-local by Def. 1, so Thm. 1 applies to all of them, and MarSea is complementary rather than competing: relational information still enters the score through $b(\gamma_{ij})$, and what changes is that the allocation is then decided jointly per key. The same applies to a more recent remedy aimed at our own motivating symptom. Ye et al. (2025) take the diagnosis that attention over-allocates to irrelevant context and subtract a second row-softmax map, scaled by a learnable λ, so that the common-mode part cancels. Each map is a function of its own row, so the difference is query-local and the attainable-set identity of Thm. 1 covers it. **Our other two instruments do not transfer to it**, and it is worth being exact about why: the weights are *signed*, so Cor. 1's realization argument, which lower-bounds a column sum by Ma from a replication pair with $a > 0$ over the non-negative orthant, has no force where a column sum can be small by cancellation; and each row sums to $1 - \lambda$ rather than 1, so Prop. 3 applies only with that constant in place of one. It is not an allocation in the sense used here, and the fidelity quantities of Sec. 3.4 are not defined on it without modification. It is a good demonstration that the phenomenon is real and worth a mechanism, and it is a natural addition to the matched-sparsity baseline group of App. I.

Transport-based attention, and Slot Attention as its closest relative to the column form. Sinkformers replace row-softmax with Sinkhorn normalization to obtain doubly stochastic attention (Sander et al., 2022), following earlier use of Sinkhorn balancing for sparse attention (Tay et al., 2020), and recent work makes it efficient (Shahbazi et al., 2025). Tay et al. (2020) is in fact a different construction — Sinkhorn is applied to a block *sorting* matrix and attention over the reordered blocks remains an ordinary row-softmax — so the group that actually constrains the attention matrix is Sander et al. (2022) and Shahbazi et al. (2025). That group identifies the marginals as the surface governing mass allocation and then sets the column marginal to *uniform*: every key emits identically. Slot Attention (Locatello et al., 2020) belongs here more squarely than its usual framing suggests, and it repays being read in this paper's orientation: its slots emit the queries and its inputs supply the keys, so slot = query and input = key. It normalizes twice. The *first* normalization is a softmax over slots, which in these indices is $\sum_i \mathrm{attn}_{ij} = 1$ for every input — **a column marginal in exactly the sense of Sec. 2, fixed uniformly at one**, and the closest published relative of the column form. The second divides by the sum over inputs, giving $\sum_j A_{ij} = 1$ per slot, which is an ordinary row normalization. Zhang et al. (2023) identify the pair as a *single* Sinkhorn iteration, and one iteration does not satisfy both marginals: the second step destroys the first, so the matrix that multiplies the values is row-stochastic and its columns are unconstrained. Two things follow, and they cut in opposite directions. The construction reaches for a column marginal, which is the right object — but it is uniform and equal to one, where MarSea's column capacity is inherited from the base normalizer and differs across keys, and the competing set is a bank of slots drawn each forward pass from one shared learned Gaussian — deliberately interchangeable, so that no slot specializes — rather than the layer's own keys, so what competes is not the data whose relation is being decided. And because what survives is the row normalization, Prop. 3 applies to the result.

MESH is the closest prior art to MarSea's capacities, on *both* margins, and the difference has to be stated precisely rather than dismissed. Zhang et al. (2023) improve the construction by changing the transport *cost*, which is the headline; but their Appendix A also *learns* the marginals — and learns both of them: "*we learn both* $a = m \cdot \mathrm{softmax}(h_a(Z))$ *and* $b = m \cdot \mathrm{softmax}(h_b(X))$ *with neural networks* h_a, h_b *shared across the* m *slots or* n *input elements respectively*", so that $\sum_i a_i = \sum_j b_j = m$. Read in the orientation above, that is a learned, non-uniform, content-predicted capacity on the key margin *and* on the query margin, each normalized to a layer total — the same objects as MarSea's inherited slice masses R_j and R_i. We had written that both leave the marginals fixed; that was wrong, and it was wrong on both margins, not one.

What is absent is the other half of the object. A capacity says how much a node may emit or absorb; it does not say *which* edges carry it, and MESH does not predict that. Their coupling is an entropy-regularized transport whose solution is *dense* — the attention matrix has no exact zeros — and the strength of the entropy term is a fixed global hyperparameter, tuned once for the model, not a quantity any head predicts per slot or per input. So there is no analogue of τ on a slice, no support to recover, no interval of Thm. 2 to land in, and nothing that could make Prop. 5 inapplicable: MESH is a full-support mechanism, so its precision on any slice is pinned at $m/|\mathcal{E}|$ by that proposition exactly as row-softmax's is, however well its capacities are learned. **That is the sizable difference, and it is a difference in the modelling object rather than in the parameterization** (Table 4).

Two smaller distinctions follow from the same place. **(i) Where the relation comes from.** Their h_a reads the input element *alone* and h_b the slot alone; the capacities are coupled across participants only through a softmax normalization, and nothing decides membership. MarSea's relation logit reads the pair *anchored on the attention score* (Eq. (8)), and the program's exclusivity, whether preset or read from the slice by $\tau_\phi^{\mathsf{K}}/\tau_\phi^{\mathsf{Q}}$, acts on a slice of fixed size, which is what makes Thm. 2's interval a target rather than a moving one (Prop. 7). **(ii) Two equalities inside a transport, against one conserved margin outside one.** Their marginals are equality constraints of a doubly stochastic problem, so both margins are pinned to prescribed totals and Prop. 3 applies with equality; MarSea's one-end forms conserve the many-end margin's total and leave the one-end unnormalized, which is what lets a one-end row carry more than a unit when several keys select it, or a many-end row spend less than a unit under softmax-one. **Causal form.** A two-marginal transport under a causal mask is generically infeasible — the last key is visible to one query, whose row must then carry that key's whole marginal — and a transport run over the full sequence leaks future rows into the column marginals of earlier ones. The B2 arm of Sec. 4 is therefore MESH made causal: one Sinkhorn over the prompt, and for every answer row its own Sinkhorn over the visible prefix, warm-started from the previous row's duals (App. I).

Query coupling without a constraint. Selective Attention (Leviathan et al., 2025) subtracts an accumulated mask from the attention logits, with $F_{ij} = \sum_{k \le i-1} S_{kj}$ built from the rectified logits of a reused head, so what query i may give key j depends on what *earlier queries* decided about j. **It is therefore not query-local by Def. 1, and Thm. 1 does not cover it** — the only attention mechanism we are aware of for which that is true, and the closest existing relative of the coupling MarSea introduces. What differs is what the coupling is *for*. F is a monotone, unbounded penalty accumulated along the prefix: it makes a key progressively less available, but it enforces nothing. No quantity is constrained, no cardinality is decided, and the row remains a softmax, so $\sum_j A_{ij} = 1$ and Prop. 3 applies verbatim — some key still retains a full unit of fan-out however aggressively the mask accumulates. Our claim is not that queries should be coupled but that the coupling should *solve a program with a marginal*, which is what turns coupling into a degree bound and a recovery guarantee. Selective Attention is evidence that the field is reaching for the first half, and we would rather say so than let a reviewer find it.

Capacity in routing, and the correction we owe expert choice. Token-choice mixture-of-experts caps how many tokens an expert accepts and drops the overflow (Lepikhin et al., 2021; Fedus et al., 2022). Expert-choice routing (Zhou et al., 2022) is the entry that deserves a mechanism-level answer rather than a motivational one, and an earlier version of this paper gave the wrong kind. There, each expert independently selects its own top-k tokens with $k = nc/e$ set by a capacity factor; the selection is over tokens, it therefore *couples* them, it is a cardinality constraint on the receiving margin, and load balance holds by construction, so there is no expert overflow to discard — unlike token choice, where overflow is dropped. (A token chosen by *no* expert simply skips the layer; that is a different matter and Zhou et al. (2022) do not claim otherwise.) So the coupling and the cardinality constraint both exist there, and rejecting the comparison on the grounds that capacity is "a systems device carrying no claim about the data" was a category error. What is absent is the part the theory is about. k is a fixed hyperparameter, uniform across experts and independent of content, and the selection is a hard top-k; their capped variant does solve an entropy-regularized program, but with both margins fixed as hyperparameters (their Sec. 3.3). So there is no content-predicted allocation to recover, no exclusivity to place, and no interval of Thm. 2 to hit. (MarSea's relation is also a top-k; what runs *on* it is a program with an interval, and the comparison of Sec. 4 against row sparsemax is the comparison of the program against a hard shape.) and the adaptive dimension in this literature is the *token*'s number of experts (Yue et al., 2025), not the expert's capacity. Note also what this means for our own framing: Cor. 1 is a statement about *query-local* schemes, and expert choice is not one, so it is not a counterexample to anything we prove. Nor does it *escape* the theorem: expert choice is not an attention mechanism, so it was never inside the theorem's universe. What it shares with Selective Attention is the *shape* of the move — decide on the receiving margin, over a set of senders at once — and neither turns that move into a guarantee.

Sparse attention, and why we need no relaxation. Sparsemax (Martins & Astudillo, 2016) and α-entmax (Peters et al., 2019) yield exact zeros, and adaptively sparse transformers learn α as a static per-head scalar (Correia et al., 2019); writing attention as a regularized argmax over the probability simplex is the framework of Niculae & Blondel (2017), generalized to other convex domains, with matching losses, by Fenchel–Young losses (Blondel et al., 2020). MarSea is an instantiation of that framework on a relation slice with an inherited capacity; **the variational form is not our contribution**. Our departure from Correia et al. (2019), and from row sparsemax as we run it (B3), is threefold: the program runs on a relation of fixed size rather than the whole row, the complement of the relation is conserved at the base normalizer's values rather than zeroed, and the exclusivity is a per-program control that may be preset or read from the slice; Cor. 3 says what a preset gives up and Sec. 4 says what the first two are worth. The usual route to a differentiable selection is instead to relax a discrete operator — a transport relaxation of top-k (Xie et al., 2020), or stochastic perturbation of a black-box discrete optimizer (Berthet et al., 2020). We need neither: at $\alpha = 2$ the program's solution is a closed-form truncation (Lem. 1), differentiable almost everywhere, with no iteration and no sampling.

Softmax disperses, which is our motivating claim by a cleaner route. Veličković et al. (2025) prove that softmax attention cannot remain sharp as the number of items grows: even for a task as simple as finding the maximum key, any learned circuit must disperse at test time, and they propose adaptive temperature as an acknowledged ad-hoc remedy. This is support rather than competition,

and it reaches the motivation more cleanly than we do — but the remedy is exactly where the two part, and the difference is this paper. Their temperature is applied to a *row* at inference to sharpen a distribution that keeps full support; MarSea's exclusivity acts on a relation slice and the program it parameterizes produces exact zeros, so what it delivers is a *support* rather than a sharper point on the simplex. Prop. 9 is the quantitative form of their qualitative claim on a slice, and Prop. 5 says why sharpening alone cannot reach $P = 1$: precision is pinned by the support, and a temperature does not change the support.

Representational separations for attention. Sanford et al. (2023) is the reference class for Thm. 1: a positive separation on sparse averaging, where attention scales logarithmically in the input size and recurrent and feedforward models polynomially, together with a negative result showing attention itself scales linearly on triple detection. Their limitations are lower bounds obtained by communication complexity. Ours separates *within* attention rather than between architectures — query-local against query-coupled, at fixed depth and width — and both the separating object (a constraint rather than a function) and the method (an exact characterization of an attainable set, with an explicit witness supplied by Assumption 1) differ. What we take from that line is the standard it sets for what a separation claim has to exhibit, not a proof technique.

Assignment, agreement, and efficiency. DETR enforces one-to-one assignment at the *loss* via Hungarian matching (Carion et al., 2020); capsule routing gates information flow by agreement between lower capsules' prediction vectors and the parent capsule's output (Sabour et al., 2017). MarSea differs from the capsule family in being non-iterative, requiring no capsule-structured layers, and not renormalizing the allocation into a distribution over receivers — a slice's total is inherited from the base normalizer, not fixed at one. *Efficiency-motivated sparsifiers* for long context (Xiao et al., 2024) reduce the support to save memory; we constrain *degree* on a relation for a correctness guarantee, and the row-sparsemax arm of Sec. 4 is the comparison that keeps support *size* and *membership* from being confused.

Position bias. The position bias of Liu et al. (2024) is a distinct phenomenon with its own architectural accounts, which trace a U-shaped influence profile over *positions* to causal masking combined with residual connections (Chowdhury, 2026; Herasimchyk et al., 2026) or with relative positional encodings (Wu et al., 2025b); Chowdhury (2026) shows the profile is present at initialization. We make no claim about that axis; see App. D.

D LIMITATIONS, CONTINUED

The limitations of Sec. 4, in full. Three seeds. With 75–225 lists per cell and $n = 3$, one weak seed moves a mean by 0.05 and nothing under that is separable; we report per-seed values (Table 10) so that the reader can see which differences are one run. **Head selection.** The eight relation heads were chosen by a criterion applied to the *trained softmax arm's* block metrics on the training distribution (App. I) — a selection step the baselines neither had nor needed. It uses no evaluation data, but it uses knowledge of how softmax behaves on the task, and we state it. **One backbone, one length, LoRA.** Everything is Qwen2.5-1.5B trained at 4K; the 8K variable-tracking sets are the only out-of-distribution length and they are saturated for every arm. **The relation's reach.** A top-k relation with $k = 8$ cannot cover a supporting span of hundreds of tokens, which is why multi-hop QA is not an evaluation here (App. I); a span-scaled relation is a design change we have not made. **The row form cannot enforce what it learns.** By Thm. 1 a row-local program can select within a row and cannot prevent two rows claiming one key; the block-gold supervision teaches the partition and column purity measures how well it held, but nothing in the row form guarantees it. The column form, which does, removes repeats at inference on raw scores; on the row-normalized score it improves the weights trained through it, but its program adds nothing at inference and it has not matched the row form's exact-set (Sec. 4). **Single-layer scope:** Thm. 1 bounds what one attention operation can guarantee, not the asymptotic expressivity of a deep query-local stack — and in a decoder the query already carries the listing history from earlier layers, which is how a row-local normalizer can avoid re-listing at all: the coordination is in the query, the exact zeros are the normalizer's. **No output-error guarantee:** Thm. 2 concerns the *support* of the allocation and does not bound $\|O_i - O_i^\star\|$ without further assumptions on the values. **Everything is scoped to the relation:** pairs outside it inherit the base normalizer's precision ceiling, and an

empty relation everywhere is a degenerate optimum *easier* to reach than any collapse in τ (below). **Fidelity needs ground truth:** precision against a relation nobody annotated is not a number, which is why the block gold exists and why a task without block gold is not a fidelity measurement.

The positional account, and why it is a different claim. Theoretical accounts trace the U-shaped influence profile over *positions* largely to architecture, while differing on what else is required. Chowdhury (2026) derives it in closed form at initialization from causal masking and residual connections, and shows it unchanged by RoPE; Wu et al. (2025b), in an attention-only model, show that the causal mask alone yields primacy and characterize its trade-off with relative positional encodings such as decay masks and RoPE, and in their experiments the full U-shape also needs training data biased toward both ends; Herasimchyk et al. (2026) treat causal masking, the residual stream, positional encodings and content as four interacting forces. We do not need to adjudicate that, and we do not: what all three supply is a statement about *where* a gold item sits and is orthogonal to ours, which is about *which* of the candidates that accompany it receive mass: the two would predict the same aggregate degradation for different reasons, which is why every table in Sec. 4 holds sequence length fixed within a table and the generator places needles at uniform random depths, so that no arm sees a systematically different position profile from another. We do not claim to explain the U-shape and MarSea is not a remedy for it.

Why independent per-target selection over-commits a source. This is the content of Prop. 2(ii), and it is worth stating in words. If M targets are alike enough with respect to one source that the same source wins at each of them, then a query-local scheme assigns it to all M — no row can know that another row has already claimed it. The failure is not that the scheme cannot select; it is that it cannot decline to be selected.

Thm. 1's hypotheses, in self-attention. The theorem fixes $\{k_j\}$ while each x_i ranges over $\mathcal{D}_i$, and it is fair to ask whether the two are jointly satisfiable when q_i and k_i are projections of one token. **Fixing the keys does not by itself collapse $\mathcal{D}_i$.** Write $q_i = W_q u_i$, $k_i = W_k u_i$ with $W_q, W_k : \mathbb{R}^d \to \mathbb{R}^{d_h}$ and u_i the vector the projections read. For W_k of full rank, fixing every key confines u_i to $u_i + \ker W_k$, of dimension $d - d_h$; and unless $\mathrm{row}(W_q) \subseteq \mathrm{row}(W_k)$, in which case W_q vanishes on $\ker W_k$ and queries are fixed outright, the restriction $W_q|_{\ker W_k}$ has rank $\min(d_h,\, d - d_h)$. So q_i still moves in a d_h-dimensional set once $d \geq 2d_h$, and under pre-LN, where u_i additionally lies on a $(d-2)$-dimensional manifold, once $d \geq 2d_h + 2$ — three or more heads at $d = Hd_h$. Two honest caveats: the rank statements are generic rather than universal, and trained weights are not a generic draw, so the row-space degeneracy is measurable and unmeasured; and rank is not conditioning — a small $\sigma_{\min}(W_q|_{\ker W_k})$ makes the required displacement large, which the pre-LN constraint bounds. **What this establishes is the negative claim, not a positive one**: the key constraint alone does not make Thm. 1 vacuous. Whether the tuple $(u_1, \ldots, u_{n_q})$ ranges over a *product* is a separate question, and it is the one Assumption 1 answers — so the division of labour is that Thm. 1 supplies the *shape* of what a query-local scheme attains, on whatever product domain it is given, and Assumption 1 with Cor. 1 supplies the *bite*. The second is the half a shrunken domain can actually threaten, and it is the hypothesis a positive result for the column form would need to be read against: whether real answer rows are near-equivalent with respect to one key is a property of the trained representation, not of the theorem.

Degeneracy, and how it is detected. Two collapses must be told apart. An empty relation *is* the base normalizer entrywise (Prop. 1(iii)) and is the degenerate optimum easiest to reach; the relation-off control of Sec. 4 is that state evaluated on purpose. $\tau \to 0^+$ on a non-empty slice is *not* — it spreads the inherited mass uniformly over the slice (Lem. 1: every prefix has $G < 1$), a slice reshaped to the wrong shape — and with τ preset it cannot occur by training. The subtler collapse is a relation of *wrong* members: a slice that holds the right number of keys but not the source, which by Cor. 2 no τ can repair. That is what the block precision and recall of the relation measure (App. I), and why they are reported from the first evaluation rather than inferred from task accuracy.

The one-end is unnormalized in the column form. A one-end row is the target of many keys' programs and is not renormalized afterwards, so its total may exceed a unit when several keys select it, and output magnitudes vary across rows within a layer more than they do under softmax. The

row form has no such asymmetry: every row's total is exactly its base total. We have not measured whether the column form's unnormalized rows cost anything downstream.

Both margins at once. Running the two forms together constrains both margins, which is the general exclusive relation of Sec. 2.1. Two things are open. The form that runs second inherits the first's output rather than the base mass, so the first form's support guarantee holds only up to what the second removes; and a joint statement — exact support on both margins — needs the target sets of the two margins to be consistent (a key's target rows are the rows whose target keys include it, and conversely), a hypothesis an earlier version of this work stated as a corollary and that we no longer claim. Neither the joint form nor that hypothesis is evaluated in this paper; every reported run uses one form per task.

Smaller items. $\alpha = 2$ **specificity.** Monotone exclusivity and Thm. 2 are proved for sparsemax, the setting we deploy; monotonicity is numerically consistent at $\alpha \in \{1.2, 1.5, 1.8, 3\}$ but unproven there. **Not invariant to the query set (column form).** The column program couples queries, so it is permutation-equivariant but not invariant to insertion or deletion of queries; the row form is. **The column form interacts with the key–value cache.** ψ_j is a function of the visible slice, so admitting a new query alters previously computed entries; Prop. 11 bounds the damage in the direction that matters and the frozen-prefix variant is fully incremental at the cost of over-admission. **A preset τ.** Cor. 3 says exactly which slices a preset fails — those with $\delta < 1/\tau$ — and the learned heads $\tau_\phi^{\mathsf{K}}, \tau_\phi^{\mathsf{Q}}$ exist in the implementation; we have not evaluated them in this round.

E FIDELITY ON THE MANY-END, IN FULL

This appendix expands Sec. 3.4: what Def. 2 measures and why, the precision–recall reading of the recovery interval, and why neither a softmax nor a larger candidate set escapes the identity of Prop. 5.

What the definition measures. The support in Def. 2 is taken *on the relation*: off it the row or column carries the base normalizer's dense entries by construction, and precision there is not a quantity the mechanism decides. Under any *normalized* allocation mass-precision and mass-recall coincide at ρ, so the two error types become definable only once exact zeros exist; this is a capability a fully supported mechanism lacks, not a vocabulary.

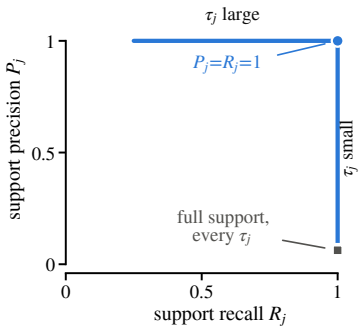

Figure 1: **The trace is predicted in full.** One slice at 400 exclusivities, $|\mathcal{E}| = 64$, $m = 4$: below the interval $R = 1$ and $P = m/k^\star$, on it $P = R = 1$, above it $P = 1$ and $R = k^\star/m$, and no interior point. Full support is the grey square, at *every* τ.

The identity is not a technicality about support. A mechanism whose output set is fixed in advance has surrendered precision as a *controllable* quantity: the scores may rank the targets perfectly while the precision is unchanged. In list retrieval, the members of an answer row's slice that are *already listed* are exactly non-targets that a full-support row cannot give zero.

The recovery interval is a precision–recall interval. By Prop. 8, $\tau \geq 1/(W + m\delta)$ is exactly the precision requirement and $\tau < 1/W$ exactly the recall requirement, with $P = R = \rho = 1$ inside; because both endpoints are exact, so is the correspondence (Fig. 1). Sweeping τ traces an $\mathbf{L}$ with its corner at $P = R = 1$ and no interior point, while a full-support mechanism does not move from $(1, m/|\mathcal{E}|)$ at any temperature: recall is not what precision is bought with. Outside the interval the failure is graded rather than abrupt (Prop. 4). **Recall is the fragile side**: a slice can lose recall from τ above its interval and from the relation excluding a target, against a single precision path (τ below the interval), so no single F-style aggregate should be reported.

Smoothing is not a way around a large candidate set. Were the slice computed by a *softmax*, holding the mass outside the target set relative to that inside at ε would take $\tau \geq \frac{1}{\delta} \log \frac{|\mathcal{E}|-m}{m\varepsilon}$ in the worst case over configurations meeting Assumption 2, with the bound tight there — growing like $\log |\mathcal{E}|$ — whereas exact recovery needs $\tau \geq 1/(W + m\delta)$, with no set size displayed (Prop. 9). Every additional non-target contributes its own small share, while a terminated tail removes the dependence outright. And because the relation is a top-k of fixed size, the maximum over the complement that defines δ is taken over a set that does not grow with the context, so Assumption 2 does not become vacuous as the context does (Prop. 7, App. F).

F THE RELATION AS A SLICE: WHAT TRANSFERS, AND WHY THE MARGIN CONDITION SURVIVES THE CONTEXT

Both programs of Sec. 2 are defined on an *index set*, and in the deployed mechanism that set is a slice of fixed size k per many-end element rather than a whole row or column. This appendix keeps two things from the derivation that produced that design: the accounting of which results transfer to a slice and how, and the argument — with its measurement — for why restricting the program to a slice whose size does not track the context is what keeps Assumption 2 from becoming vacuous as the context grows. Conditional activation is the special case $\mathcal{E}_{i\cdot} \in \{\emptyset, [n_k]\}$: a many-end element with an empty slice runs no program and keeps the base normalizer.

What transfers. Table 5 is the accounting. The pattern is that results which are statements about a sparsemax and its threshold *over an index set* carry over with the index set replaced, while results that count the set do not.

Table 5: Restriction of a program to a slice, statement by statement. Every result the paper cites is in the first two groups; the third records what an earlier two-margin form owed and the one-end forms no longer need.

Result	On a slice	Because
Verbatim, with the index set replaced by $\mathcal{E}_{i\cdot}$ or $\mathcal{E}_{\cdot j}$		
Lem. 1	unchanged	a property of sparsemax on a set
Monotone exclusivity	$\tau \to 0^+$ endpoint reads $\lvert\mathcal{E}\rvert$	it names the set size
Thm. 1, separation	unchanged	the product is over the index set
Thm. 2 and its endpoints; Cor. 2	unchanged	W, m, δ carry no set size
Prop. 8, 4, 10	unchanged	same
Cor. 3, necessity	unchanged	a comparison of intervals
Prop. 11, causal form	unchanged *if* $\mathcal{E}_{\cdot j}$ is arrival-sealed	growth must add, never revoke
Prop. 1, conservation on the many-end	stated on the slice	step 1 fixes the complement
Hypothesis changes		
Assumption 2	$\max$ taken over $\mathcal{E} \setminus T$	the only place the set size entered
Cor. 1; Prop. 2(ii)	need $\lvert\mathcal{E}_{\cdot j}\rvert \geq 2$, $M \leq \lvert\mathcal{E}_{\cdot j}\rvert$	product form at a singleton
Prop. 5, ceiling	$m/\lvert\mathcal{E}\rvert$ for a *masked* comparator	support is $\mathcal{E}$, still fixed
Prop. 9, dilution	$n \to \lvert\mathcal{E}\rvert$ *at matched domain*	it describes a comparator
Owed by a two-margin form; not needed by the one-end forms		
Prop. 3, conservation	$n_q/n_k \to \mu/n_k$, μ the in-relation mass	off-$\mathcal{E}$ mass escapes it
A row cap after a column program	—	the one-end carries no program
Joint attainment on both margins	—	the programs are never composed
Prop. 6, realizability	partition of the covered set only	construction must be re-run

Why: n_q enters the theory in exactly one place. Of the three quantities fixing Thm. 2's interval, W_j and m_j are functions of the target set alone; only δ_j looks outside it, and it looks by taking a *max*

over the complement. A max over a growing pool is non-decreasing in the pool size, so δ_j is *non-increasing in n_q whatever the score law*, and it converges to $\min_{i \in T_j} s_{ij} - \operatorname{ess\,sup}(\text{non-target law})$ — which is negative whenever a non-target can outrank the weakest target with positive probability. Past that point Assumption 2 fails outright, not because the scoring function is poor but because the assumption is a statement about a set that keeps growing.

We state it this way rather than as a divergence, and the reason is our own model of the representation. If the non-target law has unbounded support above — $\mathcal{N}(\mu_0, \sigma^2)$, say — then $\delta_j \to -\infty$ at rate $\sqrt{\ln n_q}$, since the maximum is $\mu_0 + \sigma\sqrt{2\ln(n_q - m_j)}(1 + o(1))$. But under pre-LN the vector each projection reads lies on a bounded manifold (App. D), so $q_i \cdot k_j/\sqrt{d}$ is bounded and the scores are not Gaussian in the tail: the max converges *up to* an essential supremum instead of diverging. The monotone-convergence form above is what survives that, it is all the argument needs, and it is not removed by training, which moves the law but not the monotonicity. Past that point every result in Sec. 3.3 is vacuous rather than merely weak. Restricted to $\mathcal{E}_{\cdot j}$, the max is over a set whose size does not grow with the context, and δ_j becomes n_q-free by construction. Table 6 measures both, with a scoring function that already separates non-targets by two standard deviations.

Table 6: Margin and exact recovery against candidate count, at $|\mathcal{E}_{\cdot j}| = 8$, $m_j = 3$, non-targets outside the slice drawn $\mathcal{N}(-2, 1)$; 3000 columns per row, τ_j taken inside the slice's interval. The slice margin is identical across rows by construction; the column margin is not. **Both δ_j columns are means, and the recovery columns are not a function of them**, which is the honest way to read the table: recovery is per column, and Assumption 2 fails on a growing *fraction* of columns rather than switching off at a threshold. Already at $n_q = 32$, where the mean margin is still positive, 26.5% of columns have $\delta_j \le 0$ and can be recovered at no temperature; at $n_q = 128$, where the mean has gone negative, 24.5% still satisfy the assumption, which is what the residual 14.9% is made of. The slice column is 100% at every n_q because the max defining δ_j is taken over a set that does not grow.

n_q	δ_j, full column	δ_j, on $\mathcal{E}_{\cdot j}$	recovery, full column	recovery, on $\mathcal{E}_{\cdot j}$
8	1.127	1.127	100.0%	100.0%
32	0.296	1.127	63.6%	100.0%
128	−0.321	1.127	14.9%	100.0%
512	−0.791	1.127	1.2%	100.0%
2048	−1.184	1.127	0.0%	100.0%
8192	−1.555	1.127	0.0%	100.0%

What this settles about the comparison with softmax, and what it does not. Prop. 9 sets a $\log n_q$ requirement against one with "no n_q in it at all". On the unrestricted mechanism that contrast is stated more strongly than it should be: n_q survives inside δ_j on *both* sides, so the honest residual is the logarithmic factor and not the presence or absence of n. Under restriction there are two gains and they deserve separate credit. **(i)** The $\log |\mathcal{E}_{\cdot j}|$ factor belongs to the *program* — a terminated tail rather than an exponential one — and is unaffected by the mask. **(ii)** The reduction of the domain from n_q to $|\mathcal{E}_{\cdot j}|$ belongs to the *mask*, and a masked softmax would enjoy it equally. What restriction contributes beyond either is that Assumption 2 stops being asymptotically vacuous, so (i) remains a claim about something. We regard this as the strongest argument for the generalization and the correct reading of Prop. 9 without it.

The singleton, and what a fixed k does about it. The separation theorem's force is parameterized by $|\mathcal{E}_{\cdot j}|$ and vanishes at 1: cardinality on a singleton *is* product-form, so a query-local scheme enforces it by clipping and Thm. 1 says nothing; the replication pair of Assumption 1 must also be found inside the set, giving $M \le |\mathcal{E}_{\cdot j}|$. A learned relation could shrink its way out of the theorem. The top-k relation of Sec. 2.6 fixes $|\mathcal{E}| = k$ for every many-end element that has a relation at all, so this degeneracy cannot occur by training; what a learned relation can still do is admit the *wrong* k members, which is why the relation's block precision and recall are reported rather than its size.

How the relation is parameterized, and the causal form. An earlier version decided membership by the sign of a pairwise compatibility, on the argument that a pairwise decision is never revoked as the prefix grows and so keeps Prop. 11 intact — a revocable mask breaks permanence of exclusion on 90.7% of prefixes in simulation. A top-k over a row is not pairwise: whether key j is in

$\mathcal{E}_{i\cdot}$ depends on the other keys' logits. For the **row form** this costs nothing, because the row form is row-local and needs no causal argument at all. For the **column form**, membership of a row in $\mathcal{E}_{\cdot j}$ is fixed from that row's own top-k when the row is generated, and earlier rows are never revised, so the slice is arrival-sealed and Prop. 11 applies as stated. The one change a column's member list undergoes is the retention cap of App. B, which drops the lowest-scoring member and is harmless whenever $k^\star \leq 64$. The distinction the earlier argument drew still stands and is worth keeping: *whether this key and this query stand in the relation* is decided by the relation; *which of the competing candidates wins* is decided by the program; the relation restricts a domain while the program does the constraining.

Where it helps least, and the measurement that says so. A relation of fixed size k helps where the true partner set is small and the distractors are many — list retrieval, where each answer row copies from one span and the competitors are the other values' spans. It cannot help where the true partner set is itself large: a supporting sentence of a hundred tokens against $k = 8$ leaves the relation covering a few percent of the evidence, which is what the block recall of 0.02 at every relation head measures on multi-hop QA (Sec. 4). The remedy is a relation whose size scales with the span, or a sentence-level relation, and neither is a parameter of the present design.

Numerical check of the transfer. The claims of the first group are checked directly, by running the program on a slice rather than a whole margin, in the same harness that produces Table 8; Table 7 gives the counts. The margin block is Table 6. Two of these are checks that something *fails*: the n_q/n_k bound of Prop. 3 is escaped on every trial once mass may sit off $\mathcal{E}$, and permanence of exclusion breaks on 90.7% of prefixes if the mask is allowed to reconsider. The first is weaker than it looks and we mark it so: the generator draws the mask density from $[0.02, 0.5]$, so the covered mass cannot reach n_q/n_k and the escape is arranged rather than discovered — what the row shows is that the bound is not automatic under restriction, not how far it falls. As everywhere in App. H, these establish that the statements are true of the objects the program constructs; they are not evidence about trained networks, and no mask here is learned.

Table 7: Restriction to a slice, checked. Zero violations throughout, with † as in Table 8. The fourth row is run once rather than at each n_q: the slice recovery check does not depend on n_q — the pool outside the slice never enters it — so repeating it six times would report one result as six. **That n_q-invariance is the appendix's point and it is structural, not empirical**; what varies with n_q is the *full-column* comparison, which is Table 6. The last row confirms a *negative* — that the n_q/n_k bound is escaped once mass may sit off $\mathcal{E}$; that it is escaped at all is arranged by the mask density the generator draws, so the row shows the bound is not automatic, not how far it falls. The harness separately prints the rate at which a revocable slice breaks permanence of exclusion, which the fifth row's hypothesis excludes.

Claim on a slice	Test	Violations
Thm. 2, interval on a slice	$\mathrm{supp} = T$ inside, δ over the slice	0 / 24000
Thm. 2, lower endpoint	below it: $\mathrm{supp} \supsetneq T$	0 / 12000
Thm. 2, upper endpoint	at or above it: $\mathrm{supp} \subsetneq T$	0 / 10604
Assumption 2 on a slice	exact recovery, n_q-free by construction	0 / 3000
Prop. 11, arrival-sealed slice	threshold monotone, exclusion permanent	0 / 131180
Prop. 5 under a mask†	masked softmax P vs. $m/\lvert\mathcal{E}\rvert$	0 / 4000
Prop. 3, corrected bound†	$\max_j \sum_{i \in \mathcal{E}_{\cdot j}} A_{ij} \geq \mu/n_k$	0 / 3000
Prop. 3, n_q/n_k lost	the original bound is escaped off $\mathcal{E}$	0 / 3000

G PROOFS

Every numbered claim in the paper is proved here, in the order it is used in Sec. 3. Results stated in the main text are **restated** before their proofs so this appendix can be read on its own; results the main text gives in prose are stated formally here for the first time.

Notation. Every result is a statement about a sparsemax and its threshold over an index set. The proofs below are written in the *column-slice* notation of the fan-out case — target set T_j of size m_j, margin δ_j, spread W_j, exclusivity τ_j, scores $s_{\cdot j}$ over the slice $\mathcal{E}_{\cdot j}$, allocation $p_{\cdot j}$ — and the row-slice case is the same argument with the indices exchanged; the main text's subscript-free T, m, δ, W, τ

refer to whichever slice is at hand. Throughout, $\Delta^{n-1} = \{p \in \mathbb{R}^n_{\geq 0} : \sum_i p_i = 1\}$, and for a vector z we write $z_{(1)} \geq z_{(2)} \geq \cdots$ for its entries sorted descending. The solution of a sparsemax program on a slice is $p_{ij} = [\tau_j s_{ij} - \psi_j]_+$ with ψ_j the unique threshold making the entries sum to one; existence, uniqueness, the sort-based computation and almost-everywhere differentiability are standard for sparsemax (Martins & Astudillo, 2016; Peters et al., 2019), and we use them without reproof. On a slice the mechanism's entries are $A_{ij} = (R_j + \lambda T_j)\, p_{ij}$ with $R_j + \lambda T_j > 0$ whenever the base normalizer puts positive mass on the slice, which softmax and softmax-one always do; so $\mathrm{supp}(A_{\mathcal{E}_{\cdot j}, j}) = \mathrm{supp}(p_{\cdot j})$ and every support statement about p is a support statement about A.

Lemma 1 (Prefix property)**.** *For $z \in \mathbb{R}^n$ sorted descending let $G(k) := \sum_{r \leq k}(z_{(r)} - z_{(k)})$. Then $G(k{+}1) - G(k) = k(z_{(k)} - z_{(k+1)}) \geq 0$, so G is non-decreasing; consequently $\mathrm{supp}(\mathrm{sparsemax}(z))$ is the set of the $k^\star$ largest entries with $k^\star = \max\{k : G(k) < 1\}$, and $\{k : G(k) < 1\}$ is a prefix.*

Proof of Lem. 1. Write $G(k) = \sum_{r \leq k} z_{(r)} - k z_{(k)}$. Then

$$\begin{aligned} G(k{+}1) - G(k) &= \Big(\sum_{r \leq k+1} z_{(r)} - (k{+}1) z_{(k+1)} \Big) - \Big(\sum_{r \leq k} z_{(r)} - k z_{(k)} \Big) \\ &= z_{(k+1)} - (k{+}1) z_{(k+1)} + k z_{(k)} \;=\; k\big(z_{(k)} - z_{(k+1)}\big) \;\geq\; 0, \end{aligned}$$

since the entries are sorted descending. Hence G is non-decreasing, and $\{k : G(k) < 1\}$ is a prefix of $\{1, \ldots, n\}$.

For sparsemax, $p = [z - \psi]_+$ with $\sum_i [z_i - \psi]_+ = 1$. If the support consists of the k largest entries then $\psi = \big(\sum_{r \leq k} z_{(r)} - 1\big)/k$, and $z_{(k)} > \psi$ is equivalent to $k z_{(k)} > \sum_{r \leq k} z_{(r)} - 1$, i.e. $G(k) < 1$. The support size is therefore the largest such k, and by the prefix property this is well defined. □

Monotone exclusivity. Apply Lem. 1 to $z = \tau s$. Since G is positively homogeneous of degree one, $G_{\tau s}(k) = \tau G_s(k)$ and $k^\star(\tau) = \max\{k : \tau G_s(k) < 1\}$. As $G_s(k) \geq 0$ is non-decreasing in k, the set $\{k : \tau G_s(k) < 1\}$ is nested decreasing in τ, so $k^\star(\tau)$ is non-increasing. $G_s(1) = 0$ always, so $k^\star \geq 1$; as $\tau \to 0^+$ every k qualifies and $k^\star = n_q$; as $\tau \to \infty$ only k with $G_s(k) = 0$ survive, which holds iff the top k scores are tied, giving $k^\star \to |\arg\max_i s_i|$.

Theorem 1 (restated). *Fix a key j, evaluate a query-local scheme over $\mathcal{D} = \prod_i \mathcal{D}_i$ with $\{k_j\}$ fixed, and put $\mathcal{R}_i := \{A_{ij}(x_i) : x_i \in \mathcal{D}_i\}$. Then the attainable set of fan-out columns is exactly the product, $\{A_{\cdot j}(x) : x \in \mathcal{D}\} = \prod_{i=1}^{n_q} \mathcal{R}_i$, and the scheme satisfies $A_{\cdot j} \in \mathcal{C}$ for all $x \in \mathcal{D}$ iff $\prod_i \mathcal{R}_i \subseteq \mathcal{C}$. Hence no query-local scheme enforces a non-product-form $\mathcal{C}$ except vacuously.*

Proof of Thm. 1. ($\subseteq$) Fix $x = (x_1, \ldots, x_{n_q}) \in \mathcal{D}$. By Def. 1, $A_{ij}(x) = A_{ij}(x_i)$ for each i, so the column $A_{\cdot j}(x) = \big(A_{ij}(x_i)\big)_i$ has i-th entry in $\mathcal{R}_i$; hence $A_{\cdot j}(x) \in \prod_i \mathcal{R}_i$.

($\supseteq$) Let $(a_1, \ldots, a_{n_q}) \in \prod_i \mathcal{R}_i$. For each i pick $x_i \in \mathcal{D}_i$ with $A_{ij}(x_i) = a_i$, which exists by definition of $\mathcal{R}_i$. Because $\mathcal{D}$ is the *product* $\prod_i \mathcal{D}_i$, the tuple $x = (x_1, \ldots, x_{n_q})$ is admissible, and by query-locality $A_{\cdot j}(x) = (a_1, \ldots, a_{n_q})$.

This proves (9). The stated equivalence is immediate: a universally quantified guarantee over $\mathcal{D}$ holds iff it holds on the attained set, which is exactly $\prod_i \mathcal{R}_i$. If $\mathcal{C}$ is non-product-form and $\prod_i \mathcal{R}_i \subseteq \mathcal{C}$, then $\prod_i \mathcal{R}_i$ is a product subset of $\mathcal{C}$; the guarantee is then achieved not by coupling the entries but by restricting each entry's range independently, which is what we mean by vacuous. □

Corollary 1 (restated). *Capacity $\mathcal{C}_{\mathrm{cap}} = \{x : \sum_i x_i \leq c\}$ and cardinality $\mathcal{C}_{\mathrm{card}} = \{x : |\mathrm{supp}(x)| \leq m\}$ are non-product-form for $n_q \geq 2$, so by Thm. 1 no query-local scheme enforces either, for $c > 0$ and $1 \leq m < n_q$. The failure is realized on constrained input spaces: given a replication pair (a, M), some admissible input has $\sum_i A_{ij} \geq Ma$, so capacity fails whenever $c_j < Ma$ and cardinality whenever $m < M$; with $M = n_q$, no constant budget survives a query set larger than c_j/a.*

Proof of Cor. 1. Not of product form, for $c > 0$ and $1 \le m < n_q$; at $c = 0$ or $m \ge n_q$ both sets are products and the statement is vacuous, which is why those cases are excluded. Suppose $\mathcal{C}_{\text{cap}} = \prod_i \mathcal{C}_i$ with $c > 0$ and $n_q \ge 2$. The point $c\,e_1$ (all mass on entry 1) lies in $\mathcal{C}_{\text{cap}}$, so $c \in \mathcal{C}_1$; likewise $c \in \mathcal{C}_2$. Then $c(e_1 + e_2) \in \prod_i \mathcal{C}_i$ but has sum $2c > c$, a contradiction. The same argument applied to $\mathcal{C}_{\text{card}}$ with $m < n_q$, using two points whose supports are distinct singletons, gives a product element of support size $2 > m$ when $m = 1$, and generalizes to any $m < n_q$ by taking $m + 1$ singletons.

Violation. Let I index M coordinates with $\sup \mathcal{R}_i \ge a$. For any $\epsilon > 0$ choose $x_i \in \mathcal{D}_i$ with $A_{ij}(x_i) > a - \epsilon$ for $i \in I$, and arbitrary x_i otherwise; admissible because $\mathcal{D}$ is a product. Then $\sum_i A_{ij} > M(a - \epsilon)$, and letting $\epsilon \to 0$ gives the bound $\ge Ma$. Thus $\sum_i A_{ij} > c$ once $M > c/a$. For cardinality, the same construction yields at least M strictly positive entries, so $|\operatorname{supp}(A_{\cdot j})| \ge M > m$.

Realization on constrained domains. By Assumption 1 there is an admissible input with a set $S \subseteq \{1, \dots, n_q\}$, $|S| \ge M$, on which $A_{ij} \ge a$. Since all entries are non-negative,

$$\sum_i A_{ij} \;\ge\; \sum_{i \in S} A_{ij} \;\ge\; |S|\, a \;\ge\; Ma.$$

Hence $\sum_i A_{ij} > c_j$ whenever $c_j < Ma$. The same input has at least M strictly positive entries in column j, so $|\operatorname{supp}(A_{\cdot j})| \ge M > m$ whenever $m < M$. Taking $M = n_q$ gives $\sum_i A_{ij} \ge n_q a$, which exceeds c_j once $n_q > c_j/a$. □

Proposition 3 (restated). *If $\sum_j A_{ij} = 1$ for every query, then $\max_j \sum_i A_{ij} \ge n_q/n_k$. No row-normalized scheme can bound every key's fan-out below n_q/n_k; doubly stochastic attention attains this with equality for every key.*

Proof of Prop. 3. Summing the row constraint over queries gives $\sum_{i,j} A_{ij} = \sum_i 1 = n_q$. A maximum is at least an average, so $\max_j \sum_i A_{ij} \ge \frac{1}{n_k} \sum_j \sum_i A_{ij} = n_q/n_k$. For a doubly stochastic scheme with uniform column marginal every column sums to exactly n_q/n_k by definition, attaining the bound. Since $n_q/n_k \to \infty$ as n_q grows with n_k fixed, no constant bound on all columns is possible.

The restriction to *all* columns is essential. A row-normalized scheme may bound one column — e.g. by driving $s_{ij} \to -\infty$ for a single j — whereupon the remaining columns absorb the mass. Cor. 1 excludes this for query-local schemes on different grounds, namely that the bound must hold uniformly over admissible inputs. □

Proposition 1 (restated). *For every relation $\mathcal{E}$, every $\tau > 0$ and every $\lambda \in [0, 1]$: (i) in the row form, $\sum_j A_{ij} = \sum_j A^N_{ij}$ for every query i; (ii) entries off the slice equal $(1 - \lambda)A^N_{ij}$ exactly; (iii) with $\mathcal{E} = \emptyset$ the output equals A^N entrywise; (iv) in the column form, $\sum_{i \in Q_j(t)} A_{ij} = \sum_{i \in Q_j(t)} A^N_{ij}$ for every key j, and rows are not renormalized; (v) the row form is query-local and the column form is not; (vi) on the slice, every member past the sparsemax threshold receives an exact zero.*

Proof of Prop. 1. (i) By (3), $\sum_{j \in \mathcal{E}_{i\cdot}} A_{ij} = (R_i + \lambda T_i) \sum_j p_{ij} = R_i + \lambda T_i$ since $p_{i\cdot}$ lies on the simplex, and $\sum_{j \notin \mathcal{E}_{i\cdot}} A_{ij} = (1 - \lambda)T_i$; the total is $R_i + T_i = \sum_j A^N_{ij}$. A row with an empty slice keeps $A^N_{i\cdot}$ by definition and the identity is trivial. (ii) is the second clause of (3) read off. (iii) With $\mathcal{E} = \emptyset$ every slice is empty, so no program runs on either margin and every entry is A^N_{ij}; under softmax-one, A^N is by definition the softmax-one normalizer, which is the B1 arm. (iv) The same computation as (i) along a column over $Q_j(t)$: the slice's total is $R_j + \lambda T_j$ and the complement's $(1 - \lambda)T_j$. Nothing in (7) divides by a row sum, so rows are as the column programs leave them. (v) In (3) every quantity — $A^N_{i\cdot}$, R_i, T_i, $\mathcal{E}_{i\cdot}$ (the top-k of $\ell_{i\cdot}$, a function of q_i and the keys), and $s_{i,\mathcal{E}_{i\cdot}}$ — is a function of $(x_i, \{k_j\}_j)$ alone, so the row is, which is Def. 1. In (7) the entry A_{ij} for $i \in \mathcal{E}_{\cdot j}$ is $(R_j + \lambda T_j)\,[\tau s_{ij} - \psi_j]_+$ where ψ_j is determined by $\{s_{i'j} : i' \in \mathcal{E}_{\cdot j}\}$ and R_j, T_j by $\{A^N_{i'j} : i' \in Q_j(t)\}$, both of which depend on other queries' inputs; a single admissible change to $x_{i'}$ that moves $s_{i'j}$ across the threshold changes A_{ij}, so the column form is not query-local. (vi) is

(4): $p_{ij} = [\tau s_{ij} - \psi_j]_+$ is exactly 0 for every member with $\tau s_{ij} \le \psi_j$, and A_{ij} is a positive multiple of p_{ij}. □

Proposition 6 (Every competitive selection is realizable by the column form)**.** *Let $\{T_j\}$ partition $\{1, \dots, n_q\}$ with $m_j = |T_j| \ge 1$. Then scores and a single preset exclusivity exist such that the column form gives $\operatorname{supp}(A_{\cdot j}) = T_j$ for every j simultaneously, for every $\lambda \in [0, 1]$; the construction survives any sup-norm score perturbation below $1/2$. The cardinalities may differ arbitrarily subject only to what a partition forces, $\sum_j m_j = n_q$.*

Proof of Prop. 6. Take $s_{ij} = \mathbb{1}[i \in T_j]$ and $\mathcal{E}_{\cdot j} = [n_q]$. Fix j. Since the T_j are non-empty, every target in T_j scores 1 and every other query scores 0, so $\delta_j = 1 > 0$ and, all scores within T_j being equal, $W_j = 0$. By Thm. 2 the recovery interval is $[1/m_j, \infty)$, which contains $\tau = 1$ for every j, so one preset $\tau = 1$ serves every column; by the uniqueness and permutation symmetry of the program, $p_{\cdot j}$ is uniform on T_j. By the notational remark above $\operatorname{supp}(A_{\cdot j}) = \operatorname{supp}(p_{\cdot j}) = T_j$ for every λ, since the slice mass $R_j + \lambda T_j$ is positive under either base normalizer. For robustness, let s' satisfy $\|s' - s\|_\infty < 1/2$: every target still scores above $1/2$ and every non-target below, so $\delta'_j > 0$, the top m_j entries remain T_j, and Thm. 2 applies with the perturbed constants for every τ_j in the perturbed interval. The partition is used only for $\sum_j m_j = n_q$; dropping it to arbitrary non-empty $\{T_j\}$ changes nothing in the argument, since the column form does not renormalize rows and a row selected by several keys simply carries their combined mass. □

Proposition 7 (Shrinking a relation only helps)**.** *Let $T_j \subseteq \mathcal{E}' \subseteq \mathcal{E}$ be nested slices sharing the target set. Then W_j and m_j are unchanged, and **(i)** $\delta_j(\mathcal{E}') \ge \delta_j(\mathcal{E})$; **(ii)** the relative interval width $m_j\delta_j/W_j$ is non-decreasing; **(iii)** if Assumption 2 holds on $\mathcal{E}$ it holds on $\mathcal{E}'$, and the recovery interval on $\mathcal{E}$ is* contained in *the one on $\mathcal{E}'$; **(iv)** the converse fails — Assumption 2 can hold on $\mathcal{E}'$ where it fails on $\mathcal{E}$. Restricting to a smaller relation therefore never narrows the admissible exclusivity set and can create one where none existed; in particular, a top-k relation is safe for recovery.*

Proof of Prop. 7. W_j and m_j are functions of $\{s_{ij} : i \in T_j\}$ alone and T_j is common to both sets. (i) $\delta_j(\mathcal{E}) = \min_{T_j} s - \max_{\mathcal{E} \setminus T_j} s$ and $\mathcal{E}' \setminus T_j \subseteq \mathcal{E} \setminus T_j$, so the maximum over the smaller set is no larger. (ii) Immediate from (i) with W_j, m_j fixed. (iii) If $\delta_j(\mathcal{E}) > 0$ then $\delta_j(\mathcal{E}') \ge \delta_j(\mathcal{E}) > 0$, so Assumption 2 holds on $\mathcal{E}'$; the lower endpoint $1/(W_j + m_j\delta_j)$ is decreasing in δ_j and the upper endpoint $1/W_j$ does not depend on the set, giving $[1/(W_j + m_j\delta_j(\mathcal{E})), 1/W_j) \subseteq [1/(W_j + m_j\delta_j(\mathcal{E}')), 1/W_j)$. (iv) Take $T_j = \{1\}$, $\mathcal{E}' = \{1, 2\}$, $\mathcal{E} = \{1, 2, 3\}$ with $s_1 = 1, s_2 = 0, s_3 = 2$: then $\delta_j(\mathcal{E}') = 1 > 0 > -1 = \delta_j(\mathcal{E})$. □

Proposition 2 (restated). *Fix a query-local scheme. **(i)** Selection at a* single *target is available to it: row-sparsemax at large temperature admits only $\arg\max_j s_{ij}$ at every target. **(ii)** Scope is not: for a scope bound κ_j (the requirement being $|T_j| \le \kappa_j$), a replication pair (a, M) with $\kappa_j < M$ drives $|\operatorname{supp}(A_{\cdot j})| \ge M > \kappa_j$ on some admissible input. **(iii)** Hence the non-local content is neither in isolation but their* conjunction*, which the column form satisfies by construction and the row form does not.*

Proof of Prop. 2. (i) Let $A_{i\cdot} := \operatorname{sparsemax}(\lambda\, s_{i\cdot})$ for $\lambda > 0$. This is query-local: the row depends on q_i, $\gamma_{i\cdot}$ and $\{k_j\}_j$ only, and on no $x_{i'}$ with $i' \ne i$. By monotone exclusivity applied along the row, $|\operatorname{supp}(\operatorname{sparsemax}(\lambda s_{i\cdot}))| \to |\arg\max_j s_{ij}|$ as $\lambda \to \infty$, which is 1 for scores in general position; and by Lem. 1 the surviving index is the largest, namely $\arg\max_j s_{ij}$. Choosing λ beyond the finite threshold at which the support size reaches its limit gives the claim.

(ii) By Assumption 1 there is an admissible input on which at least M queries satisfy $A_{ij} \ge a > 0$; each such query contributes a strictly positive entry to column j, so $|\operatorname{supp}(A_{\cdot j})| \ge M$. If $\kappa_j < M$ this violates $|T_j| \le \kappa_j$. The argument is the cardinality half of Cor. 1 and holds for every query-local scheme admitting a replication pair.

(iii) Immediate from (i) and (ii): the class attains target-side exclusivity, fails scope, and therefore fails their conjunction. The column form attains the conjunction: realizability is Prop. 6, and for scope the relevant quantity is a *cardinality*, which Lem. 1 supplies directly — $|\operatorname{supp}(A_{\cdot j})| =$

$|\operatorname{supp}(p_{\cdot j})| = k^\star(\tau_j)$, non-increasing in τ_j by monotone exclusivity and equal to m_j under Assumption 2 with τ_j in (10). The row form does not attain it: by Prop. 1(v) it is query-local, so (ii) applies to it verbatim. □

Theorem 2 (restated). *Let Assumption 2 hold on a slice. Then* $\operatorname{supp}(p_{\cdot j}) = T_j$ *if and only if* $\tau_j \in [\,1/(W_j + m_j\delta_j),\ 1/W_j\,)$*, with* $1/W_j := \infty$ *when* $W_j = 0$*; this interval is non-empty for every column with a positive margin.*

Proof of Thm. 2. Let $z := \tau_j s_{\cdot j}$ with $\tau_j > 0$. By Assumption 2, $\min_{i\in T_j} s_{ij} > \max_{i\notin T_j} s_{ij}$, so the m_j largest entries of z are exactly the indices in T_j. Write G for the quantity in Lem. 1 applied to z, and recall from that lemma that the support of $\operatorname{sparsemax}(z)$ is the top $k^\star$ entries with $k^\star = \max\{k : G(k) < 1\}$. It therefore suffices to show $k^\star = m_j$.

Step 1: $k^\star \geq m_j$, i.e. $G(m_j) < 1$. By definition,

$$G(m_j) \;=\; \sum_{r\leq m_j} \big(z_{(r)} - z_{(m_j)}\big) \;=\; \tau_j \sum_{i\in T_j} \Big(s_{ij} - \min_{i'\in T_j} s_{i'j}\Big) \;=\; \tau_j W_j,$$

using that the top m_j entries are T_j and that $z_{(m_j)}$ corresponds to the smallest score within T_j. Hence $G(m_j) < 1$ iff $\tau_j W_j < 1$ iff $\tau_j < 1/W_j$: an *equivalence*, not merely a sufficient condition.

Step 2: $k^\star \leq m_j$, i.e. $G(m_j{+}1) \geq 1$. The top m_j entries are T_j and $z_{(m_j+1)}$ corresponds to the largest score outside T_j, which by Assumption 2 is $\min_{i\in T_j} s_{ij} - \delta_j$. So $G(m_j{+}1)$ is not merely bounded below but *available in closed form*: writing $s_{\min} := \min_{i\in T_j} s_{ij}$,

$$G(m_j{+}1) \;=\; \sum_{r\leq m_j} \big(z_{(r)} - z_{(m_j+1)}\big) \;=\; \tau_j\Big(\sum_{i\in T_j} \big(s_{ij} - s_{\min}\big) + m_j\delta_j\Big) \;=\; \tau_j\big(W_j + m_j\delta_j\big),$$

the $r = m_j{+}1$ term vanishing. Hence $G(m_j{+}1) \geq 1$ iff $\tau_j \geq 1/(W_j{+}m_j\delta_j)$, again an equivalence.

By Lem. 1 the feasible set $\{k : G(k) < 1\}$ is a prefix, so $G(m_j{+}1) \geq 1$ implies $G(k) \geq 1$ for all $k \geq m_j{+}1$ and hence $k^\star \leq m_j$. Combining the two steps, $k^\star = m_j$ — and the support is T_j — exactly when τ_j lies in (10). Finally that interval is non-empty iff $W_j < W_j + m_j\delta_j$, i.e. iff $m_j\delta_j > 0$, which Assumption 2 and $m_j \geq 1$ give (interpreting $1/0 = \infty$).

Remark. Bounding $G(m_j{+}1) \geq m_j\tau_j\delta_j$ instead yields the sufficient interval $[1/(m_j\delta_j), 1/W_j)$, which is non-empty only when $m_j\delta_j > W_j$; that condition is an artifact of the bound and not a property of the problem. □

Proof of Cor. 2. With $m_j = 1$, $W_j = \sum_{i\in T_j}(s_{ij} - \min_{T_j} s) = 0$, so (10) reads $[1/\delta_j, \infty)$ and Thm. 2 gives exact recovery iff $\tau_j \geq 1/\delta_j$, i.e. iff $\delta_j \geq 1/\tau_j$. Inside the interval the support is the singleton T_j, so p puts mass one on it and A puts the whole slice mass $R_j + \lambda T_j$ on it, with exact zeros elsewhere by (4). If $\delta_j \leq 0$ the top entry of the slice is a non-target, so by Lem. 1 the support — a prefix of the sorted order — contains a non-target for every τ_j, and Assumption 2 fails, so no τ_j recovers T_j. As $\tau_j \to \infty$, monotone exclusivity gives $k^\star \to |\arg\max s_{\cdot j}|$, which for scores in general position is the single top non-target, so the target's entry is an exact zero; as $\tau_j \to 0^+$ every prefix has $G < 1$, $k^\star = |\mathcal{E}_{\cdot j}|$, and the allocation tends to uniform, $1/|\mathcal{E}_{\cdot j}|$ on every member including the target. □

Corollary 3 (restated). *If two keys satisfy* $W_{j_2} + m_{j_2}\delta_{j_2} \leq W_{j_1}$ *their recovery intervals are disjoint, so no single global τ achieves exact recovery for both; a content-predicted τ_j does.*

Proof of Cor. 3. Let $I_j := \big[1/(W_j + m_j\delta_j),\ 1/W_j\big)$, which by Thm. 2 is *exactly* the set of temperatures recovering key j. The hypothesis $W_{j_2} + m_{j_2}\delta_{j_2} \leq W_{j_1}$ gives $\sup I_{j_1} = 1/W_{j_1} \leq 1/(W_{j_2} + m_{j_2}\delta_{j_2}) = \inf I_{j_2}$, and since I_{j_1} is open on the right, $I_{j_1} \cap I_{j_2} = \emptyset$. A single global temperature lies in at most one of the two sets, so — because Thm. 2 is an equivalence and not merely a sufficient condition — exact recovery genuinely fails for at least one of the two keys. A per-key τ_j can be chosen in each interval separately. □

Proposition 5 (restated). *Let a mechanism assign $p_{ij} > 0$ to every pair — row-softmax, any additive bias on it, any elementwise gate with positive factors. Then for every key, every score configuration and every temperature, $P_j = m_j/n_q$ on every column and $P_i = |K_i|/n_k$ on every row. Both are independent of the scoring function, so neither improves by training, by architecture within the class, or by any temperature; both tend to* 0 *as the candidate set grows. Recall is* 1 *trivially.*

Proof of Prop. 5. If $p_{ij} > 0$ for every i then $S_j = \mathrm{supp}(p_{\cdot j}) = \{1, \dots, n_q\}$, so $|S_j| = n_q$ and $|S_j \cap T_j| = |T_j| = m_j$, giving $P_j = m_j/n_q$ by Def. 2. Neither quantity refers to the scores, so the value is unchanged by any reparameterization of the scoring function or any positive temperature; row-softmax has $p_{ij} = e^{s_{ij}} / \sum_{i'} e^{s_{i'j}} > 0$ for all finite scores, and an elementwise gate with positive factors preserves positivity. The row statement is identical with the margins exchanged. Finally $m_j/n_q \to 0$ as $n_q \to \infty$ at fixed m_j. □

Proposition 8 (The recovery interval is a precision–recall interval)**.** *The endpoints of (10) govern the two error types separately, and each is an equivalence: $\tau_j \geq 1/(W_j + m_j\delta_j)$ is exactly the **precision** requirement ($P_j = 1$ iff it holds, and below it $P_j < 1$ with $R_j = 1$) and $\tau_j < 1/W_j$ exactly the **recall** requirement ($R_j = 1$ iff it holds, and above it $R_j < 1$ with $P_j = 1$), with $P_j = R_j = \rho_j = 1$ inside. Since the interval is always non-empty, both are attainable at a single temperature on* every *column with a positive margin; what a hard column costs is not attainability but the* width $m_j\delta_j/W_j$ *of the target the head must hit.*

Proof of Prop. 8. Write G for the prefix statistic of Lem. 1 applied to $z = \tau_j s_{\cdot j}$ and recall $k^\star = \max\{k : G(k) < 1\}$, with $\mathrm{supp} =$ the top $k^\star$ entries.

(i) Step 2 of the proof of Thm. 2 computes $G(m_j{+}1) = \tau_j(W_j + m_j\delta_j)$ exactly, so $k^\star \leq m_j$ iff $\tau_j \geq 1/(W_j + m_j\delta_j)$. If instead $\tau_j < 1/(W_j + m_j\delta_j)$ then $G(m_j{+}1) < 1$, so $k^\star > m_j$ and the support strictly contains T_j; since by Assumption 2 the entries ranked beyond m_j lie outside T_j, we get $|S_j| > m_j = |S_j \cap T_j|$ and hence $P_j < 1$, while $R_j = 1$. No extremal configuration is needed: the equivalence holds for every column.

(ii) Step 1 establishes $k^\star \geq m_j$ from $G(m_j) = \tau_j W_j < 1$, i.e. $\tau_j < 1/W_j$. If $\tau_j \geq 1/W_j$ then $G(m_j) \geq 1$, so $k^\star < m_j$ and the support is a proper subset of the top m_j entries, which are exactly T_j; hence $|S_j \cap T_j| = |S_j| < m_j$, giving $R_j < 1$ with $P_j = 1$.

Combining, both equal one simultaneously iff τ_j lies in $[1/(W_j + m_j\delta_j), 1/W_j)$, which is non-empty whenever $\delta_j > 0$.

Exactness inside the interval. By Thm. 2, $S_j = \mathrm{supp}(p_{\cdot j}) = T_j$ for every τ_j in (10). Then $|S_j \cap T_j| = |T_j| = m_j = |S_j|$, so $P_j = R_j = 1$ by Def. 2. Every coordinate outside T_j is exactly zero, so $\rho_j = \sum_{i \in T_j} p_{ij} = \sum_i p_{ij} = 1$. □

Proposition 4 (restated). *Past the upper endpoint of its interval a slice sheds true members in* non-decreasing *order of strength — one at a time except that tied members shed together — with precision pinned at* 1 *throughout: for $\tau_j > 1/W_j$, $P_j = 1$ and $R_j = k(\tau_j)/m_j$ with $k(\tau_j) = \max\{k : W_j(k) < 1/\tau_j\}$, $W_j(k) := \sum_{r \leq k}(s_{(r)} - s_{(k)})$.*

Proof of Prop. 4. (i) Column. Let $\tau_j > 1/W_j$. By part (ii) of the previous proof $k^\star < m_j$, so the support is contained in the top m_j entries, which by Assumption 2 are exactly T_j; hence $S_j \subseteq T_j$, giving $P_j = 1$ and $R_j = |S_j|/m_j = k^\star/m_j$. For $k \leq m_j$ the top k entries all lie in T_j, so $G(k) = \tau_j \sum_{r \leq k}(s_{(r)} - s_{(k)}) = \tau_j W_j(k)$, and $k^\star = \max\{k : \tau_j W_j(k) < 1\} = \max\{k : W_j(k) < 1/\tau_j\}$, which is the claimed $k(\tau_j)$. Monotonicity of $W_j(k)$ in k is Lem. 1, so the surviving set is a prefix in score order and the lowest-scoring targets are shed first. □

Proposition 9 (A smooth column dilutes with the candidate set)**.** *Suppose the fan-out were computed by column-*softmax *and let Assumption 2 hold. Then the mass outside the target set relative to that inside obeys $\sum_{i \notin T_j} p_{ij} \,/\, \sum_{i \in T_j} p_{ij} \leq \frac{n_q - m_j}{m_j} e^{-\tau_j \delta_j}$, so $\tau_j \geq \frac{1}{\delta_j} \log \frac{n_q - m_j}{m_j \varepsilon}$* suffices *to hold the ratio at ε; and the bound is attained with equality — so that temperature is also* required *— whenever the non-targets tie at $\max_{i \notin T_j} s_{ij}$ and the targets at $\min_{i \in T_j} s_{ij}$. The requirement therefore grows like $\log n_q$ in the worst case over configurations meeting Assumption 2, whereas exact recovery at*

$\alpha = 2$ needs only $\tau_j \geq 1/(W_j + m_j\delta_j)$, in which n_q does not appear. App. F reads the comparison honestly: n_q survives inside δ_j on both sides, so the residual difference is the logarithmic factor.

Proof of Prop. 9. By Assumption 2, $s_{ij} \leq \max_{i' \notin T_j} s_{i'j}$ for every $i \notin T_j$ and $s_{ij} \geq \min_{i' \in T_j} s_{i'j}$ for every $i \in T_j$, and these two quantities differ by δ_j. The normalizing constant cancels in the ratio, so

$$\frac{\sum_{i \notin T_j} e^{\tau_j s_{ij}}}{\sum_{i \in T_j} e^{\tau_j s_{ij}}} \;\leq\; \frac{(n_q - m_j)\,\exp\big(\tau_j \max_{i \notin T_j} s_{ij}\big)}{m_j\,\exp\big(\tau_j \min_{i \in T_j} s_{ij}\big)} \;=\; \frac{n_q - m_j}{m_j}\, e^{-\tau_j \delta_j},$$

using $\tau_j > 0$ and the definition of δ_j, which is the stated bound. Setting the right-hand side to ε and solving for τ_j gives a *sufficient* temperature, since the criterion was reached from an upper bound. It is also necessary in the worst case: if every non-target sits at $\max_{i \notin T_j} s_{ij}$ and every target at $\min_{i \in T_j} s_{ij}$, both inequalities above hold with equality, the ratio is exactly $\frac{n_q - m_j}{m_j} e^{-\tau_j \delta_j}$, and no smaller τ_j holds it at ε. This is the configuration the $\log n_q$ claim is about; on configurations with the non-targets spread well below their maximum a smaller τ_j suffices. The comparison with Thm. 2 is immediate either way, that bound being $\tau_j \geq 1/(W_j + m_j\delta_j)$ with no dependence on n_q. □

Proposition 10 (Exactness of hierarchical selection, and fidelity under retention)**.** *Let $k^\star = |\operatorname{supp}(\operatorname{sparsemax}(\tau_j s_{\cdot j}))|$. **(i)** If $K \geq k^\star$, hierarchical top-K retention followed by one exact program on the survivors yields $p_{\cdot j}$ identical to the global solution, for any block size and merge arity. **(ii)** If Assumption 2 holds with τ_j in (10) and $K < m_j$, the surviving support is the K highest-scoring members of T_j, whence $P_j = 1$, $R_j = K/m_j$ and $\rho_j = 1$: under-retention is a* pure recall *failure, exact in rate — and invisible to ρ_j.*

Proof of Prop. 10. (i) Exactness. By Lem. 1 the support of $\operatorname{sparsemax}(z)$ is the set of the $k^\star$ largest entries of z, and the solution on that set is $p_i = z_i - \psi$ with ψ determined by those entries alone. It therefore suffices that no member of the global top-$k^\star$ is discarded at any level, since the root program then sees a superset of the global support and reproduces both $k^\star$ and ψ.

Retention is by score order within each group, so a member of the global top-$k^\star$ is discarded at some level only if more than K entries of that group outrank it; every such entry is then also globally ranked above it, hence itself a member of the global top-$k^\star$, giving more than K members of a set of size $k^\star$ and so $K < k^\star$. Contrapositively, $K \geq k^\star$ implies no member is ever discarded, at any level and for any block size or merge arity. The root group receives every survivor, and solving the program there over a superset of the global support returns the global $p_{\cdot j}$ exactly, the coordinates outside the support being zero in both.

(ii) Under-retention. Here $k^\star = m_j$, since Assumption 2 with τ_j in (10) gives $\operatorname{supp}(p_{\cdot j}) = T_j$ by Thm. 2; so $K < m_j$ means $K < k^\star$ and part (i) does not apply. What does still apply is the *retention* argument used inside (i): retention is by score order at every level, and a globally top-K entry is discarded only if more than K entries of its group outrank it, each of which is then itself globally top-K — impossible. So the survivor set is exactly the K globally highest-scoring entries.

By Assumption 2 the top m_j entries are T_j, so for $K < m_j$ the survivor set is the K highest-scoring members of T_j (ties at rank K broken arbitrarily, the conclusion being unaffected). Solving the program on that set: writing G for the statistic of Lem. 1, $k^\star = m_j$ means $G(k) < 1$ for every $k \leq m_j$, hence in particular for every $k \leq K$; and G restricted to the survivors agrees with the global G for $k \leq K$, since $G(k)$ depends only on the top k entries, which are the same in both. So the support is all K survivors. Therefore S_j is a subset of T_j of size K, giving $|S_j \cap T_j| = |S_j| = K$ and so $P_j = 1$ and $R_j = K/m_j$ by Def. 2. Finally the allocation is normalized over the survivors, all of which lie in T_j, so $\rho_j = \sum_{i \in T_j} p_{ij} = 1$. □

Proposition 11 (Prefix monotonicity: growth is edge-removing)**.** *Let $\psi_j^{(t)}$ be the Stage-1 threshold on the visible set $Q_j(t)$, at fixed τ_j. Then **(i)** $\psi_j^{(t)}$ is non-decreasing in t; **(ii)** exclusion is permanent ($p_{ij}^{(t)} = 0$ implies $p_{ij}^{(t')} = 0$ for $t' \geq t$) while admission is provisional; **(iii)** the frozen-prefix variant over-admits and never under-admits, so its error is one-sided.*

Proof of Prop. 11. (i) At $\alpha = 2$ the threshold ψ is the unique solution of $\sum_{i \in Q}[z_i - \psi]_+ = 1$, where $z = \tau_j s_{\cdot j}$ restricted to the visible set Q. Write $F_Q(\psi) := \sum_{i \in Q}[z_i - \psi]_+$. For fixed

Q, F_Q is continuous, non-increasing, and strictly decreasing wherever positive, so the solution of $F_Q(\psi) = 1$ is unique. Enlarging the set to $Q' = Q \cup \{i^\dagger\}$ adds a non-negative term pointwise, so $F_{Q'}(\psi) \geq F_Q(\psi)$ for every ψ. Evaluating at the old solution gives $F_{Q'}(\psi_j^{(t)}) \geq 1$, and since $F_{Q'}$ is non-increasing the solution of $F_{Q'}(\psi) = 1$ is at least $\psi_j^{(t)}$. Hence $\psi_j^{(t+1)} \geq \psi_j^{(t)}$, and by induction ψ_j is non-decreasing along the prefix.

(ii) $p_{ij}^{(t)} = [z_i - \psi_j^{(t)}]_+$, so $p_{ij}^{(t)} = 0$ means $z_i \leq \psi_j^{(t)}$. By (i), $\psi_j^{(t')} \geq \psi_j^{(t)} \geq z_i$ for $t' \geq t$, whence $p_{ij}^{(t')} = 0$. The converse fails: $z_i > \psi_j^{(t)}$ does not imply $z_i > \psi_j^{(t')}$, so an admitted query may be evicted.

(iii) The frozen variant assigns index i the value it had when i first became visible, i.e. at $t = i$, giving $p_{ij}^{\text{frozen}} = [z_i - \psi_j^{(i)}]_+$. Since $\psi_j^{(i)} \leq \psi_j^{(T)}$ by (i), we have $p_{ij}^{\text{frozen}} \geq p_{ij}^{(T)}$ pointwise, so $p_{ij}^{(T)} > 0 \Rightarrow p_{ij}^{\text{frozen}} > 0$ and the frozen support contains the exact one. The inclusion may be strict, so relative to the exact prefix column the frozen variant admits a superset and never drops a member: its error is one-sided in *support*, which is what (iii) asserts and what the column form's frozen-prefix emission policy relies on. We do not claim it is one-sided in *fidelity*: P_j and R_j are measured against T_j, not against the exact support, and this proposition assumes no margin condition, so an extra admitted index may be a target and precision may rise. Under Assumption 2 with τ_j in (10), where the exact support is T_j, the two coincide and the frozen variant's extra admissions are exactly false positives. □

H NUMERICAL VERIFICATION OF THE ANALYTICAL CLAIMS

What the harness is, and what it is not. The harness checks the theorems *as stated*: sparsemax at exclusivity τ on an index set with a planted target set, which is the object every result in Sec. 3 and App. G is about. It predates the deployed configuration — it was written against an earlier, composed version of the mechanism — and we have kept only the rows whose claims survive in the present paper; rows that tested results the composed version needed and the one-end programs do not (a unit cap, a second stage on the rows, a joint-attainment corollary) are removed rather than reinterpreted. The harness knows nothing about the base normalizer, about λ, or about a trained relation; the mass $(R + \lambda T)$ that a one-end program carries multiplies a sparsemax allocation and moves no support (Prop. 1(v)), so every support-level claim below is checked on the allocation p alone. None of it is evidence about real data; Sec. 4 is.

Protocol. Unless a claim requires otherwise, a slice is drawn by choosing its size uniformly from a stated range, choosing a target-set size m, and drawing scores so that Assumption 2 holds by construction: targets receive scores in $[0, 1)$ and non-targets scores below $-\delta$ for a margin δ drawn uniformly from $[0.2, 1.2)$. This plants a known ground truth with a known positive margin, which is what makes support recovery checkable rather than merely plausible. **No block discards a slice because the claim failed on it**, which is a consequence of the sharp form of Thm. 2: once the interval is known to be non-empty whenever $\delta > 0$, every block samples the interval itself. **Two rows count fewer evaluations than slices drawn, and the table shows it** (10,098 and 20,472 against 12,000 and 24,000; the same test on a relation slice, which the program also prints, runs 10,604 of 12,000): those rows test behaviour *above* the upper endpoint $1/W$, which is $+\infty$ when $W = 0$, so a slice with $m = 1$ has no such regime and contributes nothing to that row while contributing normally to every other. That is the test being undefined, not a draw being rejected, and we report the counts that were actually run rather than the counts attempted. Exact-arithmetic claims are checked exactly on integers and sets; floating-point claims use 10^{-9}. Each block seeds its own generator, so the program is deterministic and the table is reproduced exactly by running it; total runtime is under a minute on one CPU core and the only dependency is `numpy`. The last block checks App. F and is reported there.

How large is $k^\star$? The realized support size $k^\star$ is what a fixed relation size k must exceed for the top-k relation to lose nothing (App. B). Over 4000 slices per regime, with slice sizes drawn from $\{64, \ldots, 2048\}$ and scores $\mathcal{N}(0, 2^2)$ scaled by τ: when the slice is *exclusive* ($\tau \in [0.5, 6]$) the median is 1, the 99th percentile 5 and the maximum 8; in a *mixed* regime ($\tau \in [0.1, 20]$) the

Table 8: Verification of the analytical claims. Denominators are **evaluations**, not slices: a slice swept at four exclusivities contributes four. † **marks a row whose conclusion follows from how the generator builds its input**, so it can confirm an algebraic identity but cannot fail — the text below says what each of these does and does not test. ‡ **marks a row that re-expresses evaluations already counted** in the Thm. 2 sharpness rows. No slice is discarded for failing a claim; rows testing the regime above $1/W$ count only the slices for which that regime exists ($W > 0$), which is why two denominators fall short of the slices drawn. App. F is checked separately in Table 7. The harness predates the deployed configuration and checks sparsemax on an index set; it does not exercise the base normalizer, λ, or a learned relation.

Claim	Test	Violations
Lem. 1, monotonicity of G	random vectors, $n \leq 30$	0 / 5000
Lem. 1, support characterization	sparsemax support vs. $\max\{k : G(k) < 1\}$	0 / 3000
Monotone exclusivity, $\alpha = 2$	support size vs. τ sweep	0 / 2000
Monotone exclusivity, $\alpha \in \{1.2, 1.5, 1.8, 3\}$	support size vs. τ sweep	0 / 600 each
Thm. 2, sharp recovery interval	$\mathrm{supp} = T$ for $\tau \in$ (10)	0 / 24000
Thm. 2, lower endpoint sharp	τ below it: $\mathrm{supp} \supsetneq T$, so $P < 1$, $R = 1$	0 / 12000
Thm. 2, upper endpoint sharp	τ at or above it: $\mathrm{supp} \subsetneq T$, so $R < 1$, $P = 1$	0 / 10098
Cor. 2 ($m = 1$), argmax	$\tau \geq 1/\delta$: whole mass on one member, zeros elsewhere	0 / 9000
Cor. 2 ($m = 1$), continuity	$\|p(\tau') - p(\tau)\| \leq \|\tau' - \tau\| \|s\|$ across $1/\delta$	0 / 3000
Prop. 9, dilution bound	bound on random configurations	0 / 4000
Prop. 4, graded recall	$k(\tau)$ vs. prediction, $\tau > 1/W$	0 / 3000
Prop. 10(i), exact hierarchy	hierarchical top-K vs. global, $K \geq k^\star$	0 / 2000
Prop. 10(ii), retention fidelity	P, R, ρ at $K < m$	0 / 7200
Prop. 5, ceiling†	softmax P vs. $m/\|\mathcal{E}\|$	0 / 24
Prop. 11(i)–(iii), causal column form	threshold monotone; exclusion permanent; frozen over-admits	0 / 4000 ea.
Prop. 1(ii), no relocation at $\lambda = 0$	entries off the slice equal A^N exactly	0 / 14937
Prop. 1(iii), empty relation	output $= A^N$ entrywise	0 / 2000
App. F (e), $\tau \to 0$ is uniform	slice entries $\to$ mass$/\|\mathcal{E}_{\cdot j}\|$; $\neq A^N$	0 / 1992 ea.
Prop. 7(i), margin monotone	$\delta(\mathcal{E}') \geq \delta(\mathcal{E})$	0 / 6000
Prop. 7(ii), width monotone	$m\delta/W$ non-decreasing	0 / 6000
Prop. 7(iii), containment	$I(\mathcal{E}) \subseteq I(\mathcal{E}')$	0 / 4506
Cor. 1, degree bound	$\|\,\mathrm{supp}\,\| = m$, $\|\mathcal{E}_{\cdot j}\|$ growing	**0 / 10000**
Cor. 1, precision	$P = 1$, $\|\mathcal{E}_{\cdot j}\|$ growing	**0 / 10000**
Fig. 1, trace below the interval‡	$R = 1$, $P = m/k^\star$	0 / 24000
Fig. 1, trace on the interval‡	$P = R = 1$	0 / 16000
Fig. 1, trace above the interval‡	$P = 1$, $R = k^\star/m$	0 / 20472
Fig. 1, full support is a point†	softmax at $(1, m/\|\mathcal{E}\|)$, every τ	0 / 2000

maximum is 14; and on deliberately *broad* slices ($\tau \in [0.02, 1]$) the median is 4, the 99th percentile 35 and the maximum 74. An earlier version of this work reported a single figure and concluded that a fixed size in the tens is universally safe; that was too strong. The corrected statement is more useful, because it makes the choice of k an empirical question about the deployed model rather than a constant asserted in advance, and it is why the paper reports the block statistics of App. I at the deployed $k = 8$ rather than asserting that 8 suffices.

Why some rows need few trials, and why the counts are not comparable. Prop. 5 is a deterministic identity rather than a statement that could fail on a hard draw, so it is checked on a small grid of $(|\mathcal{E}|, m)$ spanning four orders of magnitude rather than on a large sample; a larger count would add no risk to it. That reasoning generalizes, which is why the table carries daggers rather than a single total: **a large denominator on a daggered row means the loop ran many times, not that the claim survived many chances to fail.** The counts are also in different units of work — Prop. 11 under a nested scope counts every prefix timestep of every sequence, where a slice swept at four exclusivities contributes four — so the rows should be read individually and not summed. The blocks with large *informative* counts are those whose hypotheses interact.

What the daggered rows do and do not test, and why we mark them. A check can only fail if the generator can produce an input on which the conclusion is false, and for a *constructive* proposition it often cannot. Prop. 5's rows evaluate $m/|\mathcal{E}| = m/|\mathcal{E}|$ on a full support; Prop. 3's restricted row is $\max \geq \mathrm{mean}$ over n_k non-negative numbers. **These rows confirm that the algebra in the proof is the algebra in the statement, which is worth having and is how two sign errors were caught; they are not independent evidence that the propositions are true, because the generator cannot produce a counterexample.** The undaggered rows are the ones that could have failed

and did not. Thm. 2's three rows place τ on both sides of two independently computed endpoints; Prop. 10(ii) reports fidelity at $K < m$, where the claim is an inequality that a bad bound would break; Prop. 11 sweeps a growing prefix. Those are where a genuine error would have surfaced, and one did.

What none of it establishes. These are not evidence about real data: every score configuration is synthetic and every ground-truth relation planted. Whether a trained network's scores satisfy Assumption 2 on the relation it learned, and how wide the interval then is, is an empirical question about the trained relation heads, and nothing in this appendix answers it. The training logs record a column-level reading: for each key column of a relation head, δ and W as in Thm. 2 computed inside the relation (δ the lowest gold score minus the highest non-gold score), and a hit when $\tau_j = 1$ lies in (10). At step $2{,}500$ the hit rate is 0.71–0.75 over the three seeds of MarSea on the softmax-one base, 0.71–0.82 on the softmax base, and 0.23–0.35 for the column form on z at $(1, 0)$, the arm whose program adds nothing at inference (Sec. 4). The margin on the answer rows, where the row form's program runs, was not measured.

I Experimental protocol in full

Everything in Sec. 4 is reproducible from the checkpoints, evaluators and configuration files named at the end of this appendix. The protocol is the same for every arm; what differs per arm is listed once, under *Arms*.

What is trained and what is not. The backbone (Qwen2.5-1.5B base: embeddings, every attention projection, every MLP, norms, output head) is frozen in bf16. Trained, in fp32 master copies: (a) LoRA adapters on the query, key, value and output projections of *all* 28 layers, rank 16, $\alpha = 32$, dropout 0.05, 4.36M parameters, identical for every arm; (b) the arm's own modules, in the patched layers only. LoRA on every layer rather than on the patched ones alone is what makes "the same decoder under the same schedule" true: the arms must differ only in the normalizer, so every arm gets the same adaptation capacity everywhere. The patched layers are the four that hold the backbone's highest-scoring retrieval heads under the retrieval-score detector of Wu et al. (2025a) (threshold 0.1), which we run on 200 single-needle prompts at 4K: $\{14, 19, 23, 24\}$ (strongest head: layer 23, head 2); every checkpoint's configuration records them, and B1's and B3's normalizers replace softmax at every head of these four layers and nowhere else. MarSea's relation head is a pair $U, V \in \mathbb{R}^{16\times 128}$ and a scalar b_0 per relation head: 32.8K trained parameters on the eight relation heads (197K allocated across all heads of the patched layers). The learned-τ heads (835K) are constructed but never called in the reported configuration, where τ is preset, and receive no gradient. B0, B1 and B3 add nothing; B2 adds its two marginal networks.

Data: the training pool. Three sources, mixed by a length-bucketed sampler and consumed in an order that is a function of the seed alone (the data cursor is part of the checkpoint, so a restart resumes in the identical order). *RULER* (Hsieh et al., 2024), generated with the backbone tokenizer at $L = 4{,}096$ from training generator seeds that never overlap the evaluation seeds $\{100, 101, 102\}$, 400 sequences per configuration and seed: multi-value needle-in-a-haystack at $Q = 1$ over $K \in \{1, 2, 4, 8, 16, 32\}$ keys and $V \in \{1, 2, 4, 8\}$ values ($K = 32$ only to $V = 4$; $K = 8$ also at $V = 16$; 24 configurations), multi-query×multi-value at $K = 8$ with $V \in \{4, 8, 16\}$ at $Q = 2$ and $V \in \{4, 8\}$ at $Q = 4$ (5), and variable tracking over $C \in \{1, 2, 4\}$ chains and $H \in \{4, 8\}$ hops (6): 35 configurations, $42{,}000$ sequences. *MuSiQue*-answerable train (Trivedi et al., 2022), $19{,}938$ questions, passages in hop order with distractors interleaved at seeded slots. *HotpotQA* distractor train (Yang et al., 2018), a seeded $20{,}000$-question subsample, ten paragraphs as shipped. The pool holds $81{,}938$ sequences; the three sources are drawn 1:1:1, and a run of $2{,}500$ steps at batch size one sees $2{,}500$ of them. Sequences are `{context}\n\nQuestion: {question}\nAnswer: {answer}<EOS>` (RULER uses its own template and answer prefix); the loss is on the answer tokens and the EOS only; batch size one per device with no padding and no packing, so the visibility mask is the plain causal mask; sequences longer than L are dropped at generation time, never truncated.

Block gold. Every training sequence and every evaluation prompt carries a *partition* of its answer rows against its prompt spans. An *owner* is a set of answer rows: on multi-value retrieval, the rows

that emit value v; with $Q > 1$ queried keys the queried key owns its V values; on QA, the hop, with its span the hop's supporting sentences (HotpotQA, from the shipped sentence-level supporting facts mapped to token offsets) or supporting paragraph (MuSiQue, which annotates paragraphs only). A *block* is the pair (owner v's rows, the token span of v's needle or sentences); in variable tracking the span is the variable's definition mention, and distractor spans form blocks with no rows. The gold says: for the rows of owner v, the relation and the attention should fall inside block v, and no other owner's rows should reach into span v. Every key has exactly one owner and an owner may hold many keys, which is the "one true partner" structure of Sec. 2.1. An earlier evaluation marked *all* m values' tokens as gold for every answer row — a many-to-many target the mechanism is not built for — and it was replaced by the partition. The block metrics, all per head and order-free:

metric	definition
block precision	share of the relation members, and of the final mass, of owner v's rows that fall inside span v
block recall	share of span v's tokens that owner v's rows relate to
column purity	for a key in span v, the share of its related generated rows that belong to owner v
cross-owner leakage	mass from owner u's rows onto span v, $u \neq v$
stale-value mass	the special case of leakage where v was emitted earlier
repeated / omitted / extra	at generation, values listed twice, never, or without being asked for

Relation heads, and how they were chosen. The relation exists on eight heads — layer 14: heads $0, 3, 4, 5$; layer 19: heads $1, 3, 5$; layer 23: head 2 — and every other head in the patched layers, and every head elsewhere, is the base normalizer. The eight were selected by one criterion applied to the *softmax arm's* block metrics on the training distribution: a head qualifies if the softmax mass its answer rows place on their own block is at least 0.5. This is a selection step that used knowledge of softmax's behaviour and that the baselines did not need; Sec. 4 states it. A head that already routes answer rows to their own span is one on which a slice of eight keys can hold the copy source; on a head that does not, the relation would be a top-k over keys the head does not use.

The relation's training. The relation logit is Eq. (8): the detached attention score plus a rank-16 bilinear head with a per-layer bias; the slice of an answer row is its top-8 by this logit, with a straight-through gate centred at the 8-th logit so that U, V, b_0 receive a gradient. Prompt rows carry no relation. For the first 600 steps of Phase B the slice is the block gold itself (warm start); thereafter the learned top-k takes over and an auxiliary binary cross-entropy on the logit against the block gold, weight 1.0 decaying to a floor of 0.2 over $1{,}000$ steps, with four hard negatives per positive drawn from the other owners' spans, keeps membership anchored. **The auxiliary loss reaches U, V and b_0 only**: the keys, queries and the anchoring score are detached inside the relation logit, so it cannot train the backbone or the adapters, and the language-model loss is the only loss the LoRA weights see in every arm. This is what licenses the relation-off decomposition below. Two design rounds preceded this: an unanchored relation head whose top-k was 98% off-block, and membership held by the block-gold loss alone; the anchored logit with the warm start is the design that kept the copy source inside the slice on 78–88$\%$ of answer rows at step $1{,}000$ and on 82–85$\%$ at step $2{,}500$ at the strongest relation head (the probe below).

Schedule. Phase A: 500 steps of the softmax arm, one checkpoint per seed, shared by every arm with that seed number (seed s of every arm starts from `phaseA_seeds.pt`). Phase B: the arm's normalizer is switched on and training continues to step $2{,}500$. AdamW, $\beta = (0.9, 0.95)$; LoRA at peak learning rate 2×10^{-4} with weight decay 0.01, arm modules at 10^{-3} (weights, decay 0.01; biases and b_0, decay 0), joining the optimizer at step 500 with fresh moments on the running schedule; one cosine over the $2{,}500$ steps with a 3% linear warm-up decaying to 10% of peak; global gradient-norm clip 1.0; 16 sequences per optimizer step per device; bf16 autocast around the forward with every patched layer's normalizer run in fp32. The three seeds differ in data order, LoRA initialization and module initialization, and in nothing else. Eight H200 GPUs, one arm-seed per GPU; B2 is the slow arm (Sinkhorn with entropy descent on every patched layer, about a minute per training step, and 11.8 s per decoded example against softmax's 4.0 s and the row form's 6.5 s); the column form trains at about $1.5\times$ the row form's step time.

Arms. **B0**: row softmax, the Phase-A normalizer continued. **B1**: softmax-one (Miller, 2023), $A_{ij} = e^{s_{ij}}/(1 + \sum_{j'} e^{s_{ij'}})$ on every head of the patched layers; rows sum to less than one and nothing is learned. **B3**: row sparsemax, i.e. α-entmax (Peters et al., 2019) at $\alpha = 2$, on the same heads (code arm id `b4`; the launch set no α, so our class default 2 applies): the program MarSea's row form runs at $\tau = 1$, but over the whole row, no relation, no conservation of the complement. **B2**: a learned-marginal transport in the form of Zhang et al. (2023), pinned to their Secs. 3–4 and App. A, with their slots as our queries and their inputs as our keys: query marginals $a = m \cdot \mathrm{softmax}_i(h_a(\mathrm{LN}(q_i)))$ and key marginals $b = m \cdot \mathrm{softmax}_j(h_b(\mathrm{LN}(k_j)))$ with m the number of query rows, cost $-S$ on the visible pairs, entropic regularization 1 (so that with uniform marginals the plan is the softmax kernel and no temperature is introduced), four MESH entropy-descent steps on the cost with straight-through gradients, and 20 log-domain Sinkhorn iterations in fp32. *Under a causal mask the two-marginal problem is generically infeasible* — the last key is visible to one query, whose row must then carry that key's whole marginal — so Sinkhorn cannot satisfy both sides; we end the iteration on the row scaling, which the value mix needs, and log the column residual $\max_j |\sum_i A_{ij} - b_j|$ per layer. Rows are not unit (a_i is learned). This is a limitation of the baseline under causal decoding, not of our implementation of it, and it is why B2 is reported as "causal MESH". **MarSea, row form**: $\tau_{\mathrm{row}} = 1$, $\lambda_{\mathrm{row}} = 0$, $k = 8$, relation on the eight heads above, on the softmax base (`kw_p18a_anchor_blocks_l0_t1.json`) and on the softmax-one base. **MarSea, column form**: the same relation head, anchor, block gold and heads, with each value token as a many-end element committing to one answer row; on the softmax base with raw-score competition (`kw_col_t1_l0.json`; seeds 0 and 1 — the third was stopped at step 750 because it could not finish before the deadline), on the softmax-one base with raw-score competition, and on the softmax-one base with the row-normalized score z at $(\tau_{\mathrm{col}}, \lambda_{\mathrm{col}}) = (0.75, 0.1)$ (`kw_col_so_rn_t075_l01.json`) and at $(1, 0)$ (`kw_col_so_rn_t1_l0.json`). *A known deviation in one ablation*: in the code, the column's tail mass T_j also counts the prompt rows' mass on key j, which is not rescaled; at $\lambda_{\mathrm{col}} = 0$ this is inert and Prop. 1(iv) holds exactly, but at $(0.75, 0.1)$ the members receive λ_{col} times the prompt mass on j and the column total is not conserved. The column form's training step is about $1.5\times$ the row form's. **Relation off**: every MarSea checkpoint is also evaluated with $\mathcal{E} = \emptyset$, which by Prop. 1(iii) is the arm's base normalizer at inference to floating-point identity — softmax, or exactly B1's softmax-one. Two arms of an earlier protocol were not run in this round: a key-only exclusivity head (which tested a learned τ_ϕ^{K} the reported configuration does not use) and softmax sparsified to MarSea's realized support size.

Evaluation sets. *Multi-value retrieval*: ten RULER sets at 4K generated from evaluation seeds $\{0, 1, 2\}$, 25 lists each, grouped by the number of values a correct answer lists: $m = 1$ ($K8V1$, $K32V1$; 50 lists), $m = 4$ ($K8V4Q1$, $K32V4$; 50), $m = 8$ ($K8V4Q2$; 25), $m = 16$ ($K8V16Q1$, $K8V4Q4$, $K8V8Q2$; 75), $m = 32$ ($K8V16Q2$, $K8V8Q4$; 50, multi-query only). Greedy decoding under a budget of $9m + 16$ tokens per list. Per list: *exact-set* (the list is correct as a set: the set of values listed equals the gold set, so an omission or an extraneous value fails the list and a repeated value does not), per-value precision (distinct gold values listed over values listed, so repeats lower it) and recall, and the counts of repeated, omitted and extraneous values; repeats are reported in their own column. These sets share the generator and the length with training and are held out by seed, not by distribution. *Repeatability*: re-generating the same checkpoint on a different pass changes a seed's exact-set by 0.02–0.04 (one to three lists), because greedy decoding on bf16 kernels is not bit-reproducible across passes and GPU-sharing states; every reported number is from one canonical scoring pass per arm. *Variable tracking*: RULER VT at 5 and 9 variables, three sets of 25 lists per cell at 4K and at 8K; the 8K sets are the only length no arm saw in training. Every number is the mean over the three seeds; the seed standard deviation is the population standard deviation of the three seed means.

The relation-off decomposition. Because $\mathcal{E} = \emptyset$ reproduces the base normalizer exactly (Prop. 1(iii)), evaluating a MarSea checkpoint with the relation switched off measures what training *through* the program did to the adapters, with the program absent at test time; on minus off on the same checkpoint is the program's inference-time contribution; and off minus the baseline with the same base (B0 or B1) is the training-time contribution. The auxiliary block loss cannot account for the training-time part, since it reaches only U, V, b_0. On the softmax base the training-time part carries the exact-set gain on every seed (Table 10); on the softmax-one base the off weights are at B1's exact-set with fewer repeats and more omissions at 32 values, and the program at inference recovers

Table 9: **Variable tracking**, exact-set at 5 and 9 variables, means over seeds (225 lists per cell). Every arm is trained at 4K; 8K is length extrapolation. Every miss is an omission and no arm repeats a value; differences are 1–4 lists. "—": not run at 8K.

arm	n	4K, 5	4K, 9	8K, 5	8K, 9
B0 softmax	3	0.996	0.991	0.996	0.973
B1 softmax-one	3	1.000	0.991	0.982	0.978
B2 causal MESH	3	0.991	0.987	—	—
B3 row sparsemax	3	0.987	0.982	0.996	0.964
MarSea (row, softmax-one base)	3	0.996	0.991	1.000	0.982
off (= B1 at inference)	3	1.000	0.996	1.000	0.982
Row form, softmax base	3	0.996	0.982	0.996	0.982
off (= B0 at inference)	3	1.000	0.978	0.996	0.982
Column, softmax, s; 1, 0	2	0.993	0.973	—	—
Column, softmax-one, s; 1, 0	3	1.000	0.987	—	—
Column, softmax-one, z; .75, .1	3	0.996	0.991	—	—

Table 10: **Per-seed values behind Table 1**: exact-set at $m = 16$ / $m = 32$, with the relation-off control of the same checkpoint in parentheses. Seed s of every arm starts from the same Phase-A checkpoint. The hybrid row runs the row-trained softmax-one-base weights through the column program at inference; its control is the row form's.

arm	seed 0	seed 1	seed 2
B0 softmax	.89 / .62	.89 / .60	.83 / .66
B1 softmax-one	.93 / .78	.92 / .74	.92 / .70
B2 causal MESH	.77 / .56	.89 / .68	.81 / .62
B3 row sparsemax	.79 / .44	.79 / .28	.79 / .44
Row form, softmax	.96 / .76 (.93 / .72)	.81 / .62 (.89 / .58)	.97 / .74 (.93 / .76)
Row form, softmax-one	.92 / .78 (.91 / .78)	.93 / .80 (.89 / .70)	.95 / .70 (.99 / .70)
Column, softmax, s; 1, 0	.85 / .82 (.93 / .84)	.87 / .78 (.91 / .70)	—
Column, softmax-one, s; 1, 0	.89 / .64 (.93 / .62)	.92 / .66 (.92 / .64)	.91 / .62 (.92 / .62)
Column, softmax-one, z; 1, 0	.91 / .66 (.95 / .72)	.91 / .74 (.95 / .68)	.96 / .76 (.97 / .74)
Column, softmax-one, z; .75, .1	.89 / .74 (.88 / .76)	.89 / .78 (.93 / .76)	.95 / .58 (.97 / .68)
Hybrid: row weights, column program	.87 / .66 (.91 / .78)	.85 / .56 (.89 / .70)	.95 / .60 (.99 / .70)

the omissions. *A hypothesis, recorded and not tested*: inside the slice, sparsemax at $\tau = 1$ gives the language-model loss a gradient that depends on the *ranking* of the top scores with exact zeros below a threshold, where softmax's gradient is spread over the tail; over 2,000 steps this may sharpen the retrieval heads' top scores in a way that survives the relation's removal. Testing it needs a run with the relation on in training and a stop-gradient through the program, which we have not done.

Configuration selection, and why the pilots are not evidence. τ, λ and k were fixed from single-seed pilots read at step 1,000 on 25 lists per set: $\tau \in \{0.25, 0.5, 0.75, 1, 2\}$ at $\lambda = 0$, and $\lambda \in \{0.1, 0.25\}$ at $\tau \in \{0.25, 1\}$, each with its own relation-off reading. $\tau = 1, \lambda = 0$ was the best pilot and the only one above its own relation-off control; that, and only that, fixed the reported configuration. The sweep was non-monotone in τ and adjacent checkpoints of one run moved by 0.2–0.3 in exact-set at $m = 32$, which is the signature of run variance rather than of a τ effect, so the pilot values are not reported as an ablation and should not be read as one. Two pilot observations are qualitative and shaped the design: at $\tau = 4$ the program is an argmax over eight and a mis-ranking inside the slice costs the copy source everything (Cor. 2, the over-sharpening the exclusivity control exists to avoid); and every $\lambda > 0$ pilot sat below its relation-off control at step 1,000, consistent with λ moving tail mass onto a slice whose ranking the relation head had not yet learned. At full length, single seeds: on the softmax base $\lambda_{\text{row}} = 0.1$ scores 0.91 / 0.68 against its relation-off control's 0.92 / 0.70 and MarSea seed 0's 0.96 / 0.76; on the softmax-one base $(\tau_{\text{row}}, \lambda_{\text{row}}) = (0.75, 0.1)$ scores 0.92 / 0.78 against its control's 0.92 / 0.66. $\lambda = 0$ is the reported configuration on both bases.

The column form: the probe, the hybrid, and the row-normalized score. A relation probe on 40 held-out retrieval examples (3,633 answer rows) reads, per relation head, how often the copy source is in the answer row's relation and how much of the value's attention mass lands on its copying row after the program. At the strongest head (layer 19, head 3) the copy source is in the relation on 0.82–0.85 of rows for the row arms and 0.74–0.80 for the raw-score column arms, and

block precision is at its ceiling for every arm (0.31–0.36 against 0.375 for a three-token value inside $k = 8$ keys), so the column arms' relation heads are not under-trained. The landed mass is where they differ: 0.72–0.74 after the row program, 0.59–0.63 after the raw-score column program, 0.70–0.71 after the column program on z at $(1, 0)$ and 0.77–0.78 at $(0.75, 0.1)$. On z at $(1, 0)$ the copy source is in the relation on 0.86–0.89 of rows, the highest of any arm. Column purity — the share of the rows admitting key j that copy it — is 0.36–0.45 for every arm; it is a property of $\mathcal{E}$, which the score change does not touch, so the landed mass, not purity, is the quantity that measures the column program's assignment. No arm lowers stale-value mass at that head below softmax's (0.11–0.12). *Hybrid*: the row-trained softmax-one-base weights run through the column program at inference score 0.890 / 0.607 against 0.929 / 0.727 with the relation off, with more repeats (0.11 / 0.53 vs 0.06 / 0.20) — a column program needs a relation head trained under it. *Why raw scores fail*: a column's members come from different rows, each row's scores carry that row's offset, and a row with uniformly high logits wins every column it joins; the row-normalized $z_{ij} = \log A^N_{ij}$ compares members by the share of their own attention they give the key. A common per-column denominator (softmax-one over the column) was considered and rejected, since it does not remove offsets in the numerators.

The column form at softer presets. On softmax-one with z at $(\tau_{\text{col}}, \lambda_{\text{col}}) = (0.75, 0.1)$ — three seeds, and the one arm the T_j deviation above affects — exact-set is $0.910 \pm .028$ / $0.700 \pm .087$ at 16 / 32 values with the relation on and $0.927 \pm .037$ / $0.733 \pm .038$ off; precision / recall $.993$ / $.993$ and $.983$ / $.979$ on, $.994$ / $.993$ and $.982$ / $.978$ off; repeated values 0.08 / 0.43 on and 0.05 / 0.37 off; omitted values 0.12 / 0.79 on and 0.10 / 0.73 off. Against raw scores at $(1, 0)$ it gains 0.06 in 32-value exact-set but adds repeats rather than removing them (0.43 vs 0.37 off); against z at $(1, 0)$ it lands more mass on the copying row (0.77–0.78 vs 0.70–0.71) with no better lists. The presets were chosen for columns whose membership is still small early in a list; since they differ from the $(1, 0)$ arm in both controls at once and carry the T_j deviation, we read the arm as a preset ablation only. Per seed: Table 10.

Multi-hop QA. MuSiQue and HotpotQA are in every arm's training mixture but are not an evaluation in this paper: a $k = 8$ relation reaches about 2% of the tokens of a supporting span of hundreds of tokens (block recall 0.02 at every relation head), so neither form can move the mass a QA answer depends on, and a QA number would test the relation's reach rather than the program. A span-scaled or sentence-level relation is the design change that would make QA a test.

Provenance. Checkpoints `runs/<arm>/<arm>_seed<s>/final.pt` for every arm; evaluators `gen_eval.py`, `read_vt.py`; relation probes `relation_probe.json`, `relation_probe_rn.json`, `relation_probe_gold.json`; result files `final25.json` (retrieval), `final25_ctrl.json` (relation off), `final25_rowcol.json` (hybrid), `final25_gold_s{0,1,2}.json` (column form on z at $(1, 0)$), `final25_vt4k.json`, `final25_vt8k.json`; launch scripts `marsea_full.sh`, `marsea_full2.sh`, `final_eval_chain.sh`; retrieval-head detector output `detector.json`; code: the anonymized supplementary material, whose README maps each configuration file to its arm. The training pool is regenerated from the named generator seeds with the backbone tokenizer.