Stage-Partitioned Learning: A Configuration Space Between Backpropagation and Local Learning

Jaepil Jeong
Cognica, Inc.
Email: jaepil@cognica.io
Date: April 25, 2026 (draft v1.2.3)

"The most beautiful experience we can have is the mysterious. It is the fundamental emotion that stands at the cradle of true art and true science."
— Albert Einstein

Abstract

Backpropagation (BP) and fully local per-layer learning are standardly treated as distinct, and usually competing, paradigms for training deep neural networks. We show that they are, in fact, two extremal points of a single parameterized family of training configurations — the **stage-partitioned learning family** $\mathcal{C}(L)$. An element of $\mathcal{C}(L)$ is a pair (π, H) specifying how the L layers of a network are grouped into K contiguous stages via a composition $\pi = (d_0, \ldots, d_{K-1})$ of L, and whether the output heads are shared across stages or allocated per-stage (H). The partition induces a stage-local loss structure with gradient detachment between stages. We prove that BP corresponds bijectively to the degenerate configuration $\pi = (L)$, and that fully local per-layer learning corresponds to the opposite extreme $\pi = (1, \ldots, 1)$ with per-stage heads. The intermediate region realizes a genuinely new family of algorithms that inherit properties from both endpoints — memory locality, modularity, biological-plausibility, and compositional inference from the local side; end-to-end representational coherence from the BP side.

The configuration space carries natural algebraic structure: $\mathcal{C}(L)$ is a bounded lattice under refinement of partitions (with BP at the minimum and fully local at the maximum), admits a forgetful functor to the poset of compositions of L, and supports a family of training operators indexed by (π, H) that reduce to BP training and local training at the respective endpoints.

A single empirical regularity — the monotonic dependence of validation bits-per-byte (BPB) on Stage 0 depth at $L = 12$, $K = 3$, $H = \mathrm{ps}$ — turns out not to be a parochial observation but the empirical signature of a structural principle we call the **root bottleneck principle**: in a chain-partitioned model without independent input access to later stages, the first-stage representation is an information bottleneck for all later stages, and therefore any scalar metric dominated by next-token likelihood is structurally biased toward increasing the depth of the first stage. A direct corollary is the **BPB collapse proposition**: BPB is not a neutral metric over $\mathcal{C}(L)$ but a BP-descending projection of it; BPB-only optimization over $\mathcal{C}(L)$ (or over its continuous relaxation) converges to the BP endpoint $\pi = (L)$. The correct formulation of dynamic configuration selection is therefore a Pareto problem over BPB, locality/modularity, and capability-matrix utilities, not a BPB minimization.

To state this rigorously, §9 relaxes $\mathcal{C}(L)$ to a continuous configuration space $\overline{\mathcal{C}}(L) = [0,1]^{L-1} \times \mathbf{H}$ via soft detachment coefficients λ_i, and §10 extends $\mathcal{C}(L)$ to an architecture-expanded space $\mathcal{C}^+(L) = (G, S, H, O, R)$ whose free axes (stage graph G, source map S, objective family O, boundary relay R) describe which structural assumptions must be broken to obtain interior configurations competitive on BPB without surrendering capability-matrix properties. Five concrete mitigations — detached input buses, stage forests, overlapping interval experts, residual (boosted) Log-OP objectives, and bandwidth-preserving multiplicative boundary gates — are identified as the natural extensions.

Two lenses from prior work are embedded in the framework. The **Bayesian evidence lens** (§7.6) reinterprets stage heads as calibrated evidence estimators (Jeong, 2026a) and stage aggregation as Logarithmic Opinion Pooling over Bayesian posteriors (Jeong, 2026b), re-grounding Paper 1's WAND, parallel composition, and dual-head findings in a probabilistic foundation. The **answer bandwidth lens** (§8.9) sharpens the root bottleneck principle: Stage 0 dominance is not only an information bottleneck but an *answer bandwidth bottleneck* (Jeong, 2026c). A boundary operator detaches gradients but, if implemented as a bounded relay, also destroys representational capacity — so the correct summary is: *detachment is a gradient operation, not an information-deletion operation*.

The prior work of Jeong (2026d) corresponds to a specific point in $\mathcal{C}(24)$, namely $((6,6,6,6), \mathrm{shared})$, and its results are special cases of the structural framework developed here.

1. Introduction

1.1 The Dichotomy and its Dissolution

The training of deep neural networks has been conducted, almost without exception, under the algorithm introduced by Rumelhart, Hinton, and Williams (1986): backpropagation. A steady stream of alternative proposals — target propagation (Lee et al., 2015), synthetic gradients (Jaderberg et al., 2017), the HSIC bottleneck (Ma et al., 2020), Forward-Forward (Hinton, 2022), greedy layerwise training (Belilovsky et al., 2019), local representation learning (Löwe et al., 2019) — have each sought to abandon or circumvent some aspect of BP's global gradient coordination. These proposals are collectively referred to as *local learning methods*, and the literature treats them as *alternatives* to BP, to be justified against BP's performance.

This framing contains an unstated premise: that BP and local learning are categorically distinct algorithms. We argue the premise is mistaken. In this paper we construct a parameterized family $\mathcal{C}(L)$ of training configurations in which BP and fully local per-layer learning appear as two extreme elements. The intermediate region is populated by configurations, first studied at scale by Jeong (2026d), that share structural features with both endpoints.

The correct framing is not "local versus global" but *which configuration in $\mathcal{C}(L)$ is appropriate for a given purpose.* Different configurations inherit different properties from the two endpoints; each is suited to different deployment, hardware, and epistemic contexts.

1.2 The Mathematical Contribution

The principal contribution of this paper is structural and mathematical.

(a) Definition of $\mathcal{C}(L)$ (§2). We formalize the stage-partitioned learning family as a set of pairs (π, H) where π is a composition of L and $H \in \{\mathrm{sh}, \mathrm{ps}\}$ is a head structure.

(b) Algebraic structure (§3). We equip $\mathcal{C}(L)$ with a natural partial order and show that it forms a bounded lattice under refinement. The minimum element is BP and the maximum element is fully local per-layer learning.

(c) The training functor (§4). We construct a family of training operators $T(\pi, H)$ indexed by $\mathcal{C}(L)$ acting on the parameter space Θ. These operators satisfy a natural compatibility with refinement: the training dynamics of a coarser configuration can be recovered as a limiting case of a finer one under appropriate decoupling.

(d) Endpoint theorems (§5, §6). We prove two bijection theorems: the training operator $T((L), \cdot)$ is BP (up to parameterization of the output head), and the training operator $T((1, \dots, 1), \mathrm{ps})$ is fully local per-layer learning in the sense of Löwe et al. (2019) and Hinton (2022).

(e) Property inheritance (§7). We introduce a formalism for tracking which properties of the endpoints descend along which direction in $\mathcal{C}(L)$ and at what rate. Properties naturally divide into *BP-descending* (representational coherence, end-to-end optimality), *local-ascending* (memory locality, biological plausibility, modularity), and *configuration-sensitive* (capability-matrix properties such as adaptive compute and specialist composition, which are maximized in the intermediate region).

(f) The root bottleneck principle (§8.7–8.8). The Stage 0 depth dominance observed empirically (§8.1–8.6) is shown to follow from a structural information-theoretic property of chain-partitioned models: the first-stage representation is a sufficient statistic for all later stages under chain forward dependency, and therefore any BPB-dominated utility is a BP-descending projection of $\mathcal{C}(L)$. This yields the **BPB collapse proposition** (§8.8): BPB-only optimization over $\mathcal{C}(L)$ or its continuous relaxation (§9) converges to the BP endpoint. Dynamic configuration selection must therefore be formulated as Pareto selection, not BPB minimization.

(g) Relaxed and expanded configuration spaces (§9, §10). We introduce a continuous relaxation $\overline{\mathcal{C}}(L) = [0,1]^{L-1} \times \mathbf{H}$ of the configuration lattice, parameterized by layer-wise gradient transmissivity $\lambda_i \in [0,1]$, in which $\mathcal{C}(L)$ appears as the set of hard vertices. This makes dynamic (training-time, differentiable) boundary selection well-defined. To escape the BPB collapse we further expand to $\mathcal{C}^+(L) = (G, S, H, O, R)$, adding a stage graph G, source map S, objective family O, and boundary relay R. Chain-partitioned $\mathcal{C}(L)$ is the sub-case $(G = \mathrm{chain}, S(k) = k-1, O = \text{full CE}, R = \mathrm{identity})$. Five structural extensions (detached input bus, stage forest, overlapping interval experts, residual Log-OP objective, and bandwidth-preserving multiplicative relay) are identified as the natural mitigations of the root bottleneck.

(h) Two probabilistic–bandwidth lenses (§7.6, §8.9). Stage outputs are reinterpreted as *calibrated evidence estimators* in the sense of Bayesian BM25 (Jeong, 2026a), and stage aggregation is reinterpreted as Logarithmic Opinion Pooling over Bayesian posteriors (Jeong, 2026b). This re-grounds Paper 1's WAND, prefix pruning, and parallel-composition findings in a principled Bayesian framework (§7.6). Complementarily, the *answer bandwidth lens* of Jeong (2026c) sharpens the root bottleneck principle: Stage 0 dominance is the BPB signature of a root bandwidth bottleneck, and bandwidth — rather than compositional question-semantics — is the decisive property at stage boundaries. The two lenses together motivate a bandwidth- and calibration-aware multi-objective utility (§10.1) and the multiplicative boundary gate of §10.5.5.

1.3 The Empirical Finding and its Structural Reading

A single empirical regularity anchors the paper's structural claims. In §8.1–8.6 we demonstrate at 286M parameters that along one particular one-parameter slice of $\mathcal{C}(12)$ — the *Stage 0 depth axis* at $K = 3$ with per-stage heads — the validation BPB is a monotonically decreasing function of the Stage 0 depth. Read as a parochial empirical lemma, this is merely evidence that $\mathcal{C}(L)$ admits non-trivial regularities. Read structurally (§8.7–8.8), it is the empirical signature of the *root bottleneck principle*: in a chain-partitioned forward graph, every later-stage representation is a function of the first-stage representation, so no later stage can recover target-relevant information discarded at the root. The Stage 0 depth axis is therefore the coordinate along which BPB pulls the configuration space back toward the BP endpoint.

This reframing has a consequence the original lemma presentation obscures. If the only selection criterion is BPB, then optimization over $\mathcal{C}(L)$ (or the relaxed $\overline{\mathcal{C}}(L)$ of §9) is *degenerate*: it prefers $\pi = (L)$, i.e., BP. This is not a failure of the framework but a prediction of the property-inheritance theory of §7 — BPB is a BP-descending property. The whole point of the interior region is that properties other than BPB (memory locality, specialist composition, WAND, prefix pruning, dual-head preservation) are configuration-sensitive or local-ascending and reach their maxima there. Dynamic boundary selection, formalized in §10.3, is therefore a *multi-objective* Pareto selection, not a BPB minimization.

A complete empirical characterization of $\mathcal{C}(L)$ and its extensions — measuring BPB, memory footprint, parallelism overhead, capability-matrix metrics, and biological fidelity at each configuration — is stated in §10.7 as the Pareto Frontier Mapping research agenda, and is left to future work.

1.4 Relation to Prior Work

This paper is formally a sequel to Jeong (2026d), which we refer to as *Paper 1*. Paper 1 studied a specific point in $\mathcal{C}(24)$, namely $((6, 6, 6, 6), \mathrm{sh})$, at the 1.3B parameter scale and established its empirical bounded-gap property and capability matrix. In the present framework, Paper 1's findings are recovered as the empirical behavior of that specific point; the present work does not supersede Paper 1 but embeds it in the larger structure. See §11 for the detailed correspondence.

1.5 Structure of the Paper

§§2–4 define the configuration space, establish its algebraic structure, and construct the training functor. §§5–6 prove the endpoint theorems. §7 characterizes property inheritance, concluding in §7.6 with a reinterpretation of stage outputs as calibrated Bayesian evidence estimators. §8 presents the empirical finding (§§8.1–8.6) and its structural reading as the root bottleneck principle and BPB collapse proposition (§§8.7–8.8), and sharpens it in §8.9 to a root *bandwidth* bottleneck using the answer-bandwidth lens of Jeong (2026c). §9 relaxes $\mathcal{C}(L)$ to a continuous configuration space $\overline{\mathcal{C}}(L)$ via soft detachment coefficients, establishing the ambient space in which dynamic boundary selection is differentiable. §10 develops the consequences: bandwidth- and calibration-aware multi-objective configuration selection (§10.1), dynamic boundary selection at training time and post-hoc (§§10.2–10.3), the architecture-expanded configuration space $\mathcal{C}^+(L)$ with five structural mitigations of the root bottleneck (§§10.4–10.5), and the expanded Pareto Frontier Mapping agenda with a concrete experimental program (§§10.6–10.7). §11 embeds Paper 1's configuration. §12 enumerates limitations. §13 concludes.

Appendix A contains proofs deferred from the main text. Appendix B reformulates $\mathcal{C}(L)$ categorically, for readers familiar with that formalism. Appendix C records empirical trajectories.

1.6 Notation

Symbol	Definition
L	Network depth
$\pi = (d_0, \dots, d_{K-1})$	Composition of L into positive parts
$K = \lvert\pi\rvert$	Stage count
$\Pi(L)$	Set of all compositions of L
ℓ_k	$\sum_{j \le k} d_j$, terminal layer index of stage k
$H \in \{\mathrm{sh}, \mathrm{ps}\}$	Head structure (shared / per-stage)
$\mathcal{C}(L)$	$\Pi(L) \times \{\mathrm{sh}, \mathrm{ps}\}$
$\preceq$	Refinement order on $\Pi(L)$
$\sqsubseteq$	Induced order on $\mathcal{C}(L)$
Θ	Parameter space of the network
$\mathrm{sg}(\cdot)$	Stop-gradient (identity in forward pass, zero in backward pass)
$T(\pi, H)$	Training operator at configuration (π, H)
$\mathcal{L}^{\pi,H}$	Stage-partitioned loss functional at (π, H)
W_{hd}	Base output head
$W_{\mathrm{hd},k}$	Per-stage output head at stage k
h_i	Hidden state after layer i
ϕ_i	Layer map: $h_i = \phi_i(h_{i-1}; \theta_i)$
BPB	Validation bits per byte (an empirical evaluation metric)

2. The Stage-Partitioned Configuration Space

Fix an integer $L \ge 1$, the depth of the network under consideration. We define the objects of study.

2.1 Compositions

Definition 2.1. A *composition* of L is a finite ordered tuple $\pi = (d_0, d_1, \dots, d_{K-1})$ with $d_k \in \mathbb{Z}_{>0}$ and $\sum_{k=0}^{K-1} d_k = L$. The integer K is called the *length* of π and is denoted $\lvert\pi\rvert$. The set of all compositions of L is denoted $\Pi(L)$.

Proposition 2.2. The cardinality of $\Pi(L)$ is 2^{L-1}.

Proof. Compositions of L are in bijection with subsets of $\{1, \dots, L-1\}$ via $\pi = (d_0, \dots, d_{K-1}) \leftrightarrow \{\ell_0, \ell_1, \dots, \ell_{K-2}\}$ (the set of internal boundary indices). Each subset of $\{1, \dots, L-1\}$ yields one composition, and the correspondence is bijective. □

Definition 2.3 (Terminal indices). For $\pi \in \Pi(L)$, define $\ell_{-1}(\pi) = 0$ and $\ell_k(\pi) = \sum_{j=0}^{k} d_j$ for $0 \le k < \lvert\pi\rvert$. Then $0 = \ell_{-1}(\pi) < \ell_0(\pi) < \cdots < \ell_{\lvert\pi\rvert-1}(\pi) = L$. Stage k of π consists of layers $\{\ell_{k-1}(\pi) + 1, \dots, \ell_k(\pi)\}$.

2.2 Head Structures

Definition 2.4. A *head structure* is an element of the two-element set $\mathbf{H} := \{\mathrm{sh}, \mathrm{ps}\}$, with sh denoting a single shared output head across all stages, and ps denoting an additional per-stage head at each stage terminus.

Head structures carry a natural partial order: $\mathrm{sh} \le \mathrm{ps}$, reflecting that per-stage heads are a strict refinement of shared heads (adding additional parameters per stage).

2.3 The Configuration Space

Definition 2.5 (Configuration Space). The *stage-partitioned configuration space* at depth L is

$$\mathcal{C}(L) \;=\; \Pi(L) \times \mathbf{H}. \tag{1}$$

An element $(\pi, H) \in \mathcal{C}(L)$ is called a *configuration*. The first projection $\mathcal{C}(L) \to \Pi(L)$ is denoted proj_π; the second proj_H.

Proposition 2.6. $|\mathcal{C}(L)| = 2^L$.

Proof. Immediate from Proposition 2.2 and $|\mathbf{H}| = 2$. □

Remark 2.7. The configuration space $\mathcal{C}(L)$ is combinatorial and finite. For a typical deep network with $L \in [12, 100]$, the cardinality $|\mathcal{C}(L)| = 2^L$ is large but finite. The exploration of this space by exhaustive experimentation is infeasible; we regard it as the fundamental object of study, and seek to understand its algebraic and functional structure in the abstract.

2.4 Network Structure and Parameter Space

We fix a network architecture consisting of L layers $\phi_1, \dots, \phi_L$, with $\phi_i : \mathcal{H} \times \Theta_i \to \mathcal{H}$ mapping a hidden state and parameter to a new hidden state. Here $\mathcal{H}$ is the hidden-state space (typically $\mathbb{R}^d$ for hidden dimension d), and Θ_i is the parameter space of layer i. The total hidden-layer parameter space is $\Theta^{\mathrm{hid}} = \prod_{i=1}^{L} \Theta_i$.

A base output head is a map $W_{\mathrm{hd}} : \mathcal{H} \to \mathbb{R}^{|V|}$ (with V the vocabulary); per-stage heads are auxiliary maps $W_{\mathrm{hd},k}$ of the same type, one per stage. We denote the collected output-head parameters under configuration (π, H) by

$$\Theta^{\mathrm{hd}}(\pi, H) = \begin{cases} \{W_{\mathrm{hd}}\} & H = \mathrm{sh}, \\ \{W_{\mathrm{hd}}\} \cup \{W_{\mathrm{hd},k}\}_{k=0}^{|\pi|-1} & H = \mathrm{ps}. \end{cases} \tag{2}$$

The full parameter space is $\Theta(\pi, H) = \Theta^{\mathrm{hid}} \times \Theta^{\mathrm{hd}}(\pi, H)$.

Remark 2.8. The hidden-layer parameter space Θ^{hid} does not depend on (π, H); only the output-head parameters do. This is by design: the stage partition and head structure are *training-time* choices, not architectural ones.

---

3. Algebraic Structure of $\mathcal{C}(L)$

We equip $\mathcal{C}(L)$ with a partial order and establish its lattice structure.

3.1 Refinement Order on Compositions

Definition 3.1 (Refinement). Let $\pi, \pi' \in \Pi(L)$. We say π' *refines* π, written $\pi \preceq \pi'$, if every boundary of π is a boundary of π'; equivalently, the set of terminal indices $\{\ell_0(\pi), \dots, \ell_{|\pi|-2}(\pi)\}$ is a subset of $\{\ell_0(\pi'), \dots, \ell_{|\pi'|-2}(\pi')\}$. (The terminal index $\ell_{|\pi|-1}(\pi) = L$ is always shared.)

Proposition 3.2. $(\Pi(L), \preceq)$ is a bounded lattice with:

- *Minimum* $\hat{0} = (L)$ (the trivial composition with one stage);
- *Maximum* $\hat{1} = (1, 1, \dots, 1)$ (the finest composition with L stages);
- *Join* $\pi \vee \pi'$: the composition whose boundaries are the union of those of π and π';
- *Meet* $\pi \wedge \pi'$: the composition whose boundaries are the intersection of those of π and π'.

Proof. The bijection $\Pi(L) \leftrightarrow 2^{\{1,\dots,L-1\}}$ of Proposition 2.2 is order-preserving under refinement: $\pi \preceq \pi'$ iff the corresponding boundary set of π is a subset of that of π'. The subset lattice on $\{1, \dots, L-1\}$ is a Boolean lattice, hence bounded and with joins/meets given by union and intersection. □

Corollary 3.3. $(\Pi(L), \preceq)$ is isomorphic to the Boolean lattice $(\mathcal{P}(\{1, \dots, L-1\}), \subseteq)$, of order 2^{L-1}.

3.2 Order on Head Structures

Definition 3.4. The order on $\mathbf{H}$ is $\mathrm{sh} \leq \mathrm{ps}$. This makes $(\mathbf{H}, \leq)$ a two-element chain.

3.3 The Induced Order on $\mathcal{C}(L)$

Definition 3.5. On $\mathcal{C}(L) = \Pi(L) \times \mathbf{H}$ we define the product order: $(\pi, H) \sqsubseteq (\pi', H')$ iff $\pi \preceq \pi'$ and $H \leq H'$.

Proposition 3.6. $(\mathcal{C}(L), \sqsubseteq)$ is a bounded lattice with:

- *Minimum* $((L), \mathrm{sh})$, which we will show in §5 is (up to the gauge of Remark 4.8) identified with backpropagation;
- *Maximum* $((1, \dots, 1), \mathrm{ps})$, which we will show in §6 is identified with fully local per-layer learning;
- Joins and meets computed coordinatewise from the lattice structure on $\Pi(L)$ and $\mathbf{H}$.

Proof. Immediate from the fact that product of two bounded lattices is a bounded lattice. □

3.4 Atoms and Coatoms

Definition 3.7. An *atom* of $\mathcal{C}(L)$ is a minimal non-minimum element. A *coatom* is a maximal non-maximum element.

Proposition 3.8. The atoms of $\mathcal{C}(L)$ are:
(i) Configurations $((d_0, d_1), \mathrm{sh})$ with $d_0 + d_1 = L$ (single stage boundary, shared head); there are $L - 1$ such.
(ii) The configuration $((L), \mathrm{ps})$ (no stage boundary, per-stage head added).

The coatoms are:
(iii) Configurations (π^*, ps) where π^* is obtained from $(1, \dots, 1)$ by merging one adjacent pair; there are $L - 1$ such.
(iv) The configuration $((1, \dots, 1), \mathrm{sh})$ (fully partitioned, shared head).

Proof. Immediate from the product lattice structure. □

Remark 3.9. The two kinds of atoms (stage-boundary atoms and the head-structure atom) reflect the two orthogonal directions along which the configuration space can be "refined" from BP: adding a stage boundary, or adding per-stage heads. This orthogonality is a central structural feature.

3.5 Summary of Structural Results

$\mathcal{C}(L)$ is a bounded lattice of cardinality 2^L, with BP (identified in §5) at the minimum, fully local per-layer learning (identified in §6) at the maximum, and L atoms reflecting the two orthogonal modes of partition refinement. The Hasse diagram of $\mathcal{C}(L)$ thus organizes the entire space of stage-partitioned training configurations into a canonical lattice structure.

---

4. The Training Functor

We now associate to each configuration $(\pi, H) \in \mathcal{C}(L)$ a training operator on the parameter space Θ. This associates the combinatorial object $\mathcal{C}(L)$ with concrete training dynamics.

4.1 The Forward Map

Given an input $x \in \mathcal{X}$ and parameters $\theta = (\theta_1, \dots, \theta_L, W_{\mathrm{hd}}, \{W_{\mathrm{hd},k}\}_k) \in \Theta(\pi, H)$, the forward pass computes hidden states

$$h_0 = \mathrm{Embed}(x), \quad h_i = \phi_i(h_{i-1}; \theta_i) \text{ for } 1 \le i \le L. \tag{3}$$

These hidden states are *independent of* (π, H): the forward values are determined by the architecture and parameters, not by the partition.

4.2 Stage Termini and Logits

Definition 4.1 (Stage Logit). For $(\pi, H) \in \mathcal{C}(L)$ and hidden states $h_0, \dots, h_L$ as above, the *stage-k logit* is

$$z_k(\pi, H) = \begin{cases} W_{\mathrm{hd}} \cdot h_{\ell_k(\pi)} & H = \mathrm{sh}, \\ (W_{\mathrm{hd}} + W_{\mathrm{hd},k}) \cdot h_{\ell_k(\pi)} & H = \mathrm{ps}, \end{cases} \tag{4}$$

for $0 \le k < |\pi|$.

4.3 The Stage-Partitioned Loss

Definition 4.2 (Stage-Partitioned Loss Functional). Given a loss function $\mathrm{CE}(\cdot, y)$ (cross-entropy with target y) and a configuration (π, H), the *stage-partitioned loss* is

$$\mathcal{L}^{\pi,H}(\theta; x, y) = \sum_{k=0}^{|\pi|-1} \mathrm{CE}\big(z_k(\pi, H), y\big). \tag{5}$$

Remark 4.3. The loss functional is a sum of per-stage cross-entropies. It is well-defined on $\Theta(\pi, H)$; its values depend on π (which determines the set of termini) and H (which determines the logit form).

4.4 Gradient Flow with Detachment

The key ingredient is the *stop-gradient operation*. We define it operationally.

Definition 4.4 (Stop-Gradient). The operator $\mathrm{sg} : \mathcal{H} \to \mathcal{H}$ is defined by:

- *Forward pass:* $\mathrm{sg}(h) = h$ as a value (identity);
- *Backward pass:* the pull-back sg^* of sg is zero, i.e., gradients do not propagate through sg.

Formally, sg is a section of the forward map but is not a morphism in the category of differentiable maps; it is an operator that respects forward evaluation but collapses the cotangent action to zero.

Definition 4.5 (Stage-Partitioned Training Operator). For a configuration (π, H), the *stage-partitioned training operator* $T(\pi, H)$: $\Theta(\pi, H) \times \mathcal{X} \times \mathcal{Y} \to T\Theta(\pi, H)^*$ is defined as the gradient of the modified loss

$$\tilde{\mathcal{L}}^{\pi,H}(\theta; x, y) = \sum_{k=0}^{|\pi|-1} \mathrm{CE}\big(\tilde{z}_k(\pi, H), y\big) \tag{6}$$

where $\tilde{z}_k$ is obtained by replacing each stage boundary's hidden state input with its stop-gradient image:

$$\tilde{z}_k(\pi, H) = W_{\mathrm{hd},k}^{(H)} \cdot \mathrm{forward\_through\_stage}\big(k; \mathrm{sg}(h_{\ell_{k-1}(\pi)})\big). \tag{7}$$

Here $W_{\mathrm{hd},k}^{(H)}$ is either the shared head (if $H = \mathrm{sh}$) or the sum of shared and per-stage head (if $H = \mathrm{ps}$), and $\mathrm{forward\_through\_stage}(k; u)$ denotes the forward pass through the layers of stage k starting from input u.

Remark 4.6 (Why modified loss). The modified loss $\tilde{\mathcal{L}}^{\pi,H}$ has the same forward value as $\mathcal{L}^{\pi,H}$ (since sg is the identity in the forward pass), but a different gradient structure: the backward pass through sg is zero, so gradients from stage k's loss are not transmitted to stages $0, \dots, k-1$. This is the formal definition of *gradient detachment between stages*, which is the characteristic feature of stage-partitioned learning.

4.5 Gradient Locality

Proposition 4.7 (Stage-Local Gradient Support). Let θ_k^{hid} denote the parameters of the hidden layers of stage k (i.e., $\theta_{\ell_{k-1}(\pi)+1}, \dots, \theta_{\ell_k(\pi)}$). Then for any (π, H),

$$\frac{\partial \tilde{\mathcal{L}}^{\pi,H}}{\partial \theta_k^{\mathrm{hid}}} = \frac{\partial \tilde{\mathcal{L}}_k^{\pi,H}}{\partial \theta_k^{\mathrm{hid}}}, \tag{8}$$

where $\tilde{\mathcal{L}}_k^{\pi,H}$ is the k-th summand in the definition of $\tilde{\mathcal{L}}^{\pi,H}$.

Proof. By construction, the j-th summand for $j < k$ does not depend on θ_k^{hid} because the j-th stage's forward pass terminates at layer $\ell_j(\pi) < \ell_{k-1}(\pi) + 1$. The j-th summand for $j > k$ depends on θ_k^{hid} only through the hidden state $h_{\ell_k(\pi)}$, but this dependency is routed through $\mathrm{sg}(h_{\ell_k(\pi)})$ in the modified loss, whose backward pass is zero. Hence only the k-th summand contributes gradient to θ_k^{hid}. □

Proposition 4.8 (Head Gradient Structure). Under shared head ($H = \mathrm{sh}$):

$$\frac{\partial \tilde{\mathcal{L}}^{\pi,\mathrm{sh}}}{\partial W_{\mathrm{hd}}} = \sum_{k=0}^{|\pi|-1} \frac{\partial \tilde{\mathcal{L}}_k^{\pi,\mathrm{sh}}}{\partial W_{\mathrm{hd}}}. \tag{9}$$

Under per-stage head ($H = \mathrm{ps}$):

$$\frac{\partial \tilde{\mathcal{L}}^{\pi,\mathrm{ps}}}{\partial W_{\mathrm{hd},k}} = \frac{\partial \tilde{\mathcal{L}}_k^{\pi,\mathrm{ps}}}{\partial W_{\mathrm{hd},k}}, \tag{10}$$

and

$$\frac{\partial \tilde{\mathcal{L}}^{\pi,\mathrm{ps}}}{\partial W_{\mathrm{hd}}} = \sum_{k=0}^{|\pi|-1} \frac{\partial \tilde{\mathcal{L}}_k^{\pi,\mathrm{ps}}}{\partial W_{\mathrm{hd}}}. \tag{11}$$

Proof. Direct calculation from Definition 4.5. The base head W_{hd} contributes to every stage's logit and hence accumulates gradients across all stages. Per-stage heads $W_{\mathrm{hd},k}$ contribute only to stage k's logit. □

4.6 The Training Functor as a Structure-Preserving Map

Theorem 4.9 (Training Functoriality). Let $(\pi, H) \sqsubseteq (\pi', H')$ be a refinement in $\mathcal{C}(L)$. Let $\iota : \Theta(\pi, H) \hookrightarrow \Theta(\pi', H')$ be the natural inclusion (the parameter space of a coarser configuration embeds into that of a finer one by default-initialization of the additional head parameters). Then, for any (x, y):

(i) Forward compatibility. The forward values of $\mathcal{L}^{\pi,H}$ are recovered from those of $\mathcal{L}^{\pi',H'}$ by summing contributions from stages k' of π' that lie within a single stage k of π, and discarding contributions from per-stage heads not present in H.

(ii) Gradient refinement. The training operator $T(\pi, H)$ is *not* in general equal to the pullback of $T(\pi', H')$ along ι: additional stage boundaries introduce additional stop-gradient operations that zero gradients the coarser configuration does not zero. This is the essential functional difference between configurations.

Proof sketch. Part (i) is a direct computation: the forward sum over stages of π' can be rewritten as a sum over stages of π of inner sums over the stages of π' contained therein. Part (ii) follows from the fact that each internal boundary of π' (not present in π) introduces a stop-gradient operator that was not present in $T(\pi, H)$. □

Remark 4.10 (Interpretation). Theorem 4.9 formalizes the sense in which refining a configuration *strictly alters* the training dynamics. Moving from BP ($\pi = (L)$) to a non-trivial partition introduces stop-gradients that prevent certain gradient flows; moving from shared to per-stage head adds new parameters with independent gradient structure. These are genuine changes to the training operator, not merely to the loss value.

In particular, the training operator T is *not* a functor in the strict categorical sense on a category with morphisms being refinements — it is an indexed family of operators with an explicit refinement relation. We return to the categorical formulation in Appendix B.

---

5. Backpropagation as the Minimum Element

We now establish the first endpoint theorem: BP training is the training operator at $((L), \mathrm{sh})$.

5.1 Statement

Theorem 5.1 (BP-Endpoint Theorem). Let $(\pi, H) = ((L), \mathrm{sh})$. Then the training operator $T((L), \mathrm{sh})$ coincides with standard BP training, in the sense that for any parameters $\theta \in \Theta^{\mathrm{hid}} \times \{W_{\mathrm{hd}}\}$ and any input-target pair (x, y):

$$\tilde{\mathcal{L}}^{(L),\mathrm{sh}}(\theta; x, y) = \mathrm{CE}(W_{\mathrm{hd}} \cdot h_L, y) =: \mathcal{L}^{\mathrm{BP}}(\theta; x, y), \tag{12}$$

and for every parameter component θ_j,

$$\frac{\partial \tilde{\mathcal{L}}^{(L),\mathrm{sh}}}{\partial \theta_j} = \frac{\partial \mathcal{L}^{\mathrm{BP}}}{\partial \theta_j}. \tag{13}$$

5.2 Proof

Under $\pi = (L)$, the length of the composition is $|\pi| = 1$: there is a single stage, from layer 1 to layer L. The terminal index is $\ell_0((L)) = L$, and there is no earlier boundary.

Loss equality. The stage-partitioned loss consists of a single summand:

$$\tilde{\mathcal{L}}^{(L),\mathrm{sh}} = \mathrm{CE}(\tilde{z}_0, y). \tag{14}$$

Since there is no earlier boundary ($\ell_{-1}((L)) = 0$), the forward pass through stage 0 starts from $h_0 = \mathrm{Embed}(x)$ directly (no sg is applied at the stage's input). Hence $\tilde{z}_0 = W_{\mathrm{hd}} \cdot h_L$, identical to the BP output. Therefore $\tilde{\mathcal{L}}^{(L),\mathrm{sh}} = \mathcal{L}^{\mathrm{BP}}$ as values.

Gradient equality. The modified loss $\tilde{\mathcal{L}}^{(L),\mathrm{sh}}$ contains no sg operators: by definition the sg appears at stage boundaries, and there is only a single stage with no internal boundary. Hence the computational graph of $\tilde{\mathcal{L}}^{(L),\mathrm{sh}}$ is identical to that of $\mathcal{L}^{\mathrm{BP}}$. Reverse-mode differentiation produces identical gradient values at every parameter component.

This completes the proof. □

5.3 Extension to Per-Stage Head at $K = 1$

Proposition 5.2. Under $(\pi, H) = ((L), \mathrm{ps})$, the training operator is equivalent to BP with an overparameterized output head: letting $W' = W_{\mathrm{hd}} + W_{\mathrm{hd},0}$, the gradient on W' under BP equals the sum of the gradients on W_{hd} and $W_{\mathrm{hd},0}$ under $T((L), \mathrm{ps})$. The hidden-parameter gradients are identical to those of standard BP.

Proof. See Appendix A.1. Briefly: the forward logit at the single stage is $z_0 = (W_{\mathrm{hd}} + W_{\mathrm{hd},0})h_L = W'h_L$. The loss is $\mathrm{CE}(W'h_L, y)$, BP's loss for head W'. Both W_{hd} and $W_{\mathrm{hd},0}$ receive the same gradient, which equals the BP gradient on W'; their sum W' evolves consistently with BP. Hidden-parameter gradients are unchanged. □

Remark 5.3 (Gauge Symmetry). The per-stage head parameterization at $K = 1$ carries a *gauge symmetry*: the transformation $(W_{\mathrm{hd}}, W_{\mathrm{hd},0}) \mapsto (W_{\mathrm{hd}} - \Delta, W_{\mathrm{hd},0} + \Delta)$ for any Δ leaves all forward and gradient values invariant. The gauge-invariant quantity is the sum $W' = W_{\mathrm{hd}} + W_{\mathrm{hd},0}$. This redundancy is removed by picking a gauge (e.g., $W_{\mathrm{hd},0} = 0$, which recovers $((L), \mathrm{sh})$ exactly). Thus $(L), \mathrm{sh})$ and $((L), \mathrm{ps})$ are gauge-equivalent and both correspond to BP.

Corollary 5.4. The entire slice $\{((L), H) : H \in \mathbf{H}\}$ of $\mathcal{C}(L)$ is a single gauge orbit, corresponding to BP. The "minimum element" of $\mathcal{C}(L)$ in the BP sense can be taken to be $((L), \mathrm{sh})$, with $((L), \mathrm{ps})$ identified.

5.4 Interpretation

The BP-endpoint theorem establishes that backpropagation is not an algorithmic object external to $\mathcal{C}(L)$ — it is literally an element of the configuration space, occupying the lattice minimum. The algorithm commonly called "BP" is recovered from the general framework by taking the trivial composition.

---

6. Local Learning as the Maximum Element

The dual endpoint: fully local per-layer learning is the training operator at $((1, \dots, 1), \mathrm{ps})$.

6.1 Statement

Theorem 6.1 (Local Learning Endpoint Theorem). Let $(\pi, H) = ((1, 1, \dots, 1), \mathrm{ps})$ with $|\pi| = L$. Then the training operator $T((1, \dots, 1), \mathrm{ps})$ is *fully local per-layer learning*: each layer i has its own output head $W_{\mathrm{hd},i-1}$ (indexed by stage $k = i - 1$) and its own loss summand, and the gradient on the hidden parameters of layer i depends only on the per-layer loss at layer i.

6.2 Proof

Under $\pi = (1, 1, \dots, 1)$, there are L stages, each consisting of a single layer. The terminal index of stage k is $\ell_k((1, \dots, 1)) = k + 1$.

Local loss structure. The stage-partitioned loss is

$$\tilde{\mathcal{L}}^{(1,\dots,1),\mathrm{ps}} = \sum_{k=0}^{L-1} \mathrm{CE}\Big((W_{\mathrm{hd}} + W_{\mathrm{hd},k}) \cdot \tilde{h}_{k+1}^{(k)}, y\Big), \tag{15}$$

where $\tilde{h}_{k+1}^{(k)} = \phi_{k+1}(\mathrm{sg}(h_k); \theta_{k+1})$. Each stage's forward pass begins from the stop-gradient of the previous layer's hidden state.

Gradient locality. By Proposition 4.7, the gradient on θ_{k+1} (the parameters of layer $k + 1$) is

$$\frac{\partial \tilde{\mathcal{L}}^{(1,\dots,1),\mathrm{ps}}}{\partial \theta_{k+1}} = \frac{\partial \tilde{\mathcal{L}}_k^{(1,\dots,1),\mathrm{ps}}}{\partial \theta_{k+1}}, \tag{16}$$

depending only on the k-th summand. The k-th summand is the loss on the $(k + 1)$-th layer's output alone.

Identification with prior local-learning proposals. In the formulation of Löwe et al. (2019) and the Forward-Forward algorithm of Hinton (2022), each layer has an independent learning signal derived from its own output. The present formulation is exactly this: the k-th summand is a per-layer cross-entropy loss and the gradient on layer $k+1$'s parameters derives solely from this summand. The only distinction is the use of the base head W_{hd} in addition to per-layer heads $W_{\mathrm{hd},k}$; as in Corollary 5.4, one can gauge-fix $W_{\mathrm{hd}} = 0$ to recover exactly the pure per-layer formulation.

This completes the proof. □

6.3 Alternative Endpoint Forms

Remark 6.2. The fully local endpoint admits two gauge-equivalent forms:

- $((1, \dots, 1), \mathrm{ps})$ with general W_{hd};
- $((1, \dots, 1), \mathrm{ps})$ with $W_{\mathrm{hd}} = 0$, recovering pure per-layer learning.

The element $((1, \dots, 1), \mathrm{sh})$ — fully partitioned with shared head — is distinct: it lacks per-layer heads, and although each layer has its own local loss, all losses use the same single output head, which creates nontrivial coupling through the head's gradient. This configuration is an atom-of-the-coatom and interpolates between the two pure endpoints.

6.4 The Symmetric Endpoints

The two endpoint theorems, §5 and §6, establish:

- The *minimum* element of $\mathcal{C}(L)$, $((L), \mathrm{sh})$, corresponds to BP.
- The *maximum* element of $\mathcal{C}(L)$, $((1, \dots, 1), \mathrm{ps})$, corresponds to fully local per-layer learning.

The intermediate region $\mathcal{C}(L) \setminus \{\hat{0}, \hat{1}\}$ is populated by configurations that share properties with both endpoints. This is the genuinely new family. We next characterize how properties of the endpoints descend into the interior.

---

7. Property Inheritance Along the Configuration Space

Each configuration $(\pi, H) \in \mathcal{C}(L)$ inherits a bundle of properties from the two endpoints in systematically varying proportions. We distinguish three kinds of property.

7.1 BP-Descending Properties

These are properties strongest at $\hat{0} = ((L), \mathrm{sh})$ and that weaken (in a sense to be formalized) as configurations move toward the maximum.

Definition 7.1. A property $P : \mathcal{C}(L) \to \mathbb{R}$ is *BP-descending* if P is monotonically non-increasing under refinement, i.e., $(\pi, H) \sqsubseteq (\pi', H') \implies P(\pi, H) \geq P(\pi', H')$.

Principal examples:

(P1) End-to-end representational coherence. Formally: the mutual information $I(h_L; y \mid \theta)$ between the network's terminal hidden state and the target, under the training operator $T(\pi, H)$, at an optimum. BP maximizes this quantity; stage boundaries sever some of the coherence.

(P2) Gradient flow depth. The maximum depth through which a single gradient step propagates. At BP this is L; under partition π it is $\max_k d_k$.

(P3) Effective optimization target alignment. The degree to which the training objective aligns with the ultimate evaluation objective (typically, the final prediction's quality). BP optimizes the ultimate target directly; stage-partitioned configurations optimize a *sum of proxies*, which may diverge from the ultimate target.

7.2 Local-Ascending Properties

These are properties strongest at $\hat{1} = ((1, \dots, 1), \mathrm{ps})$ and weaken toward the minimum.

Definition 7.2. A property P is *local-ascending* if P is monotonically non-decreasing under refinement.

Principal examples:

(P4) Memory locality of gradients. The activation memory required to compute gradients grows with the maximum stage depth. At BP, all L activations must be retained; at fully local, only per-layer activations are needed. Formally: $\mathrm{ActMem}(\pi, H) = O(\max_k d_k)$.

(P5) Pipeline parallelism granularity. The number of disjoint chunks that can be trained on separate devices without gradient communication within a forward pass. At BP this is 1; under partition π, this is $|\pi|$.

(P6) Biological plausibility. An informal but structural property: biological neural circuits do not implement weight transport or synchronized global gradients (Grossberg, 1987; Lillicrap et al., 2016). Configurations with finer partitions require less global coordination and are closer to biological feasibility. A formalization (e.g., via the "global coordination index" of Whittington and Bogacz, 2019) places BP at maximum coordination cost and fully local at zero.

(P7) Modular update discipline. The ability to update one stage's parameters without retraining others. At BP, a parameter change in any layer propagates through the entire gradient. At fully local, each layer's parameters are updated purely by its own local loss; changes do not propagate in the backward pass.

(P8) Continual and online learning compatibility. The capacity to add new data or new tasks without forgetting past training. Formally tied to the modularity of the update: finer partitions yield better compatibility, because each stage's update is stage-local and does not globally disturb earlier-learned representations.

7.3 Configuration-Sensitive Properties

These are properties that are *maximal in the interior* of $\mathcal{C}(L)$, vanishing at both endpoints.

Definition 7.3. A property P is *configuration-sensitive* if P attains its maximum at some interior point $(\pi, H) \in \mathcal{C}(L) \setminus \{\hat{0}, \hat{1}\}$.

Principal examples:

(P9) Capability matrix. The cluster of architectural abilities established in Paper 1 — adaptive per-token compute (WAND; Broder et al., 2003 applied as in Paper 1 §5.5), prefix pruning (Paper 1 §5.2), parallel stage composition (Paper 1 §5.3), dual-head SFT preservation (Paper 1 §6.5), and multi-specialist composition (Paper 1 §10.2). These require *at least two non-trivial stages* to operate: a prefix exit cannot exist at $K = 1$, and specialist composition cannot exist without independent stage heads. They also require *non-trivial stages*: a stage of depth one with per-stage head supports the capabilities only vestigially, as the single layer cannot by itself learn a substantive specialist. Thus the capability properties are maximal in the region where $K \geq 2$ and $\min_k d_k \geq d_{\min}$ for some threshold $d_{\min}$ — empirically of order 2–6 (see §8 and Paper 1 §5).

(P10) Parallel composition gain. The gain achieved by aggregating multiple stages' predictions in log-space (Paper 1 §5.3) requires (a) at least two stages and (b) substantive diversity between stages' predictions. At $K = 1$ there is no composition; at fully local with one-layer stages, diversity is constrained by layer capacity.

(P11) Specialist capacity. The capacity of a stage to hold a non-trivial task specialization (e.g., a chat specialist, a math specialist). This requires the stage to have enough depth to encode specialization. Single-layer stages ($d_k = 1$) can support only minor output-space adjustments; deeper stages can encode substantive specialist knowledge. This is configuration-sensitive in both the partition and the depth distribution.

7.4 The Inheritance Structure

Theorem 7.4 (Property Inheritance). The set of properties partitioning into BP-descending, local-ascending, and configuration-sensitive induces a decomposition of the utility of each configuration. Formally: let $U : \mathcal{C}(L) \to \mathbb{R}$ be a utility functional. If $U = \sum_i w_i P_i$ with properties P_i of mixed types, then U is:

- Monotonically non-increasing under refinement if all w_i are BP-descending;
- Monotonically non-decreasing under refinement if all w_i are local-ascending;
- Of mixed monotonicity in general — which is the typical case.

A deployment-specific utility weights properties according to the deployment's requirements. The optimal configuration is the $\sqsubseteq$-maximum of U over $\mathcal{C}(L)$, which lies in the interior for typical utility functionals.

Proof. The first two statements are immediate from Definitions 7.1 and 7.2. The third is a consequence of the fact that typical utility functionals weight both BP-descending properties (task performance) and local-ascending properties (memory, modularity, biological fidelity) positively. □

Remark 7.5. Theorem 7.4 is the structural justification for the framework. It says that no single configuration in $\mathcal{C}(L)$ is universally optimal: the optimum depends on the utility. For deployment contexts where capability matrix properties matter, the optimum is interior. For contexts where only task BPB matters, the optimum is at the minimum (BP). This is the correct reframing of "local versus global": not as a debate about which is better, but as a selection problem within a unified space.

7.5 The Capability Region of $\mathcal{C}(L)$

Definition 7.6. The *capability region* $\mathcal{C}_{\text{cap}}(L) \subseteq \mathcal{C}(L)$ is the subset of configurations where the capability matrix properties (P9–P11) are non-vestigial, i.e., where:

- $|\pi| \geq 2$ (at least two stages, enabling stage-level abilities);
- $\min_k d_k \geq d_{\min}$ (each stage has substantive depth, enabling specialist capacity and adequate diversity).

The threshold $d_{\min}$ depends on the specific capability and the model scale. Paper 1's configuration $((6,6,6,6), \text{sh})$ at $L = 24$ lies well inside $\mathcal{C}_{\text{cap}}(24)$.

Remark 7.7. The endpoints $\hat{0}$ and $\hat{1}$ both lie *outside* $\mathcal{C}_{\text{cap}}(L)$: BP has $|\pi| = 1$; fully local has $\min_k d_k = 1$. The capability region is an interior zone, and its boundary is defined by the two conditions above. The framework clarifies why Paper 1's $((6,6,6,6), \text{sh})$ was a reasonable choice: it is a central point of $\mathcal{C}_{\text{cap}}(24)$.

7.6 Stage Outputs as Calibrated Evidence: A Bayesian Interpretation

The framework so far treats stage outputs as per-stage auxiliary classifiers summed in logit space. Two prior works of the present author admit a stronger reading: the sum is not an ad-hoc ensemble but a Bayesian evidence combination over calibrated local posteriors. We summarize the relevant facts and situate them within $\mathcal{C}(L)$.

Calibrated evidence estimators (from Bayesian BM25). Jeong (2026a) shows that an unbounded relevance score $s \in \mathbb{R}$ can be mapped into a calibrated probability $p = \sigma(\alpha(s - \mu))$ via a sigmoid likelihood model, and that this transformation (i) preserves the strict monotonicity required for ranking and (ii) preserves the upper-bound properties required for WAND / BMW safe pruning (Broder et al., 2003). The framework is general: any monotone score-to-probability map with well-behaved tails inherits these properties.

Applied to stage-partitioned learning: each stage logit z_k is a score whose softmax-calibrated counterpart $\sigma_k(y) = \text{softmax}(z_k)_y$ is a per-stage posterior estimate $p_k(y \mid x)$. Stage heads are *not merely auxiliary classifiers*; they are calibrated evidence estimators over the target. This is the minimal reading that makes the Log-OP aggregation of §4 and Paper 1 a principled inference rule rather than an ensemble heuristic.

Log-OP / PoE as Bayesian evidence combination (from Jeong, 2026b). Jeong (2026b) derives that the correct combination of multiple calibrated probability signals is geometric (log-odds) aggregation, not arithmetic averaging, and shows that the resulting composition is formally isomorphic to a feedforward neural computation. For softmax experts with logits z_k, this recovers the Log-OP / PoE rule

$$p_{\text{PoE}}(y \mid x) = \text{softmax}\Big(\textstyle\sum_k z_k\Big) \tag{17}$$

and identifies it as the normative aggregator of independent evidence — not an algorithmic choice. The same derivation reads attention as a *context-dependent Logarithmic Opinion Pool*, in which the pool weights are input-dependent posteriors over experts.

For $\mathcal{C}(L)$, this has three direct consequences:

1. **Stage aggregation is normative, not heuristic.** The training operator's stage-sum $\sum_k z_k$ is the Log-OP over stage-local calibrated posteriors; Paper 1's $+2.4$-logit parallel-composition gain (F4 in §11.2) is the evidence-composition instance of this rule.
2. **WAND pruning inherits a probabilistic safety guarantee.** Because stage outputs admit monotone upper bounds under softmax calibration, Paper 1's WAND acceleration (F2) is a *probabilistically safe* pruning, not just an empirically effective one — exactly the property established in the Bayesian BM25 framework.
3. **Depth is recursive Bayesian inference.** Jeong (2026b) reads depth as a chain of recursive marginalizations, in which each layer constructs the evidence for the next. In a chain-partitioned model, later stages operate on the evidence that the first stage constructed — the probabilistic counterpart of the information-theoretic bottleneck of §8.7.

Proposition 7.8 (Stage aggregation as Log-OP). Under Assumption 8.4 and with per-stage softmax heads, the full-model posterior $p_{\text{full}}(y \mid x) = \text{softmax}(\sum_k z_k)$ is the Logarithmic Opinion Pool of the stage-local posteriors $\{p_k(y \mid x)\}_{k=0}^{|\pi|-1}$.

Proof. By Definition 2.1–2.2 of Paper 1 and direct computation: the geometric mean of softmax posteriors is proportional to the softmax of the sum of logits, which is p_{full}. □

Remark 7.9 (On question-sequencing claims). Jeong (2026b) also conjectured a compositional *question-sequencing* principle in which ordering activations by their probabilistic semantics would improve performance. Jeong (2026c) subsequently refuted this compositional prediction experimentally: the dominant variable is representational capacity, not question semantics. The descriptive reading — that activations answer different probabilistic questions — remains useful as vocabulary, but it does not by itself determine performance. We adopt the descriptive reading in the present work and defer to §8.9 for the capacity-theoretic version.

Remark 7.10 (What the Bayesian lens does and does not imply). The Bayesian reading anchors stage aggregation, WAND safety, and parallel composition. It does not automatically imply bandwidth preservation at a boundary — calibrated probabilities live in $[0, 1]$, which is itself a bounded range that, if placed inside a hidden pathway, destroys answer bandwidth (§8.9). Calibration and bandwidth are distinct properties; both must be tracked.

---

8. Stage 0 Depth Dominance and the Root Bottleneck

This section presents the paper's empirical content together with its structural reading. §§8.1–8.6 report the one-parameter BPB slice of $\mathcal{C}(12)$ as originally measured. §§8.7–8.8 prove that the monotonicity is not a parochial fact about this slice but a structural property of chain-partitioned models: the first-stage representation is an information bottleneck for all later stages, so BPB is a BP-descending projection of $\mathcal{C}(L)$.

8.1 The Empirical Lemma

Lemma 8.1 (Stage 0 Depth Dominance). Let $L = 12$, $K = 3$, $H = \text{ps}$. Consider the one-parameter family of configurations

$$\mathcal{S} = \{(\pi, \text{ps}) \in \mathcal{C}(12) : |\pi| = 3\} \subset \mathcal{C}(12). \tag{18}$$

There is a well-defined *Stage 0 depth map* $d_0 : \mathcal{S} \to \{1, 2, \dots, 10\}$ sending (π, ps) to $\pi[0]$. Under a fixed training regime (specified in §8.2), the validation BPB is monotonically non-increasing in d_0 across the tested range $d_0 \in \{2, 3, 4, 5, 6, 8, 9, 10\}$.

Proof. Empirical; see §8.3 for the measurement. □

8.2 Experimental Setup

- **Architecture.** Decoder-only Transformer, $L = 12$, hidden $d = 768$, 12 heads.
- **Tokenizer and data.** ClimbMix, sequence length 1024.
- **Training.** Chinchilla $r = 5.63$, $2{,}365$ steps, uniform Log-OP aggregation ($\text{poe\_alpha} = 0$).
- **Parameter counts.** Per-stage configurations at 362M (base 286M + 3×25.2M per-stage heads). BP baseline at 286M.
- **Hardware.** $4 \times \text{A100 } 80\text{GB}$.
- **Seeds.** Single seed per configuration (a critical limitation; see §12).

Configurations tested and their Stage 0 depths:

Partition π	d_0	d_0/L
$(2, 5, 5)$	2	17%
$(3, 5, 4)$	3	25%
$(4, 4, 4)$	4	33%
$(5, 4, 3)$	5	42%
$(6, 3, 3)$	6	50%
$(8, 2, 2)$	8	67%
$(9, 1, 2)$	9	75%
$(9, 2, 1)$	9	75%
$(10, 1, 1)$	10	83%

8.3 Results

Final validation BPB at step 2365:

d_0	π	**Final BPB**
2	$(2, 5, 5)$	0.9663
3	$(3, 5, 4)$	0.9514
4	$(4, 4, 4)$	0.9340
5	$(5, 4, 3)$	0.9262
6	$(6, 3, 3)$	0.9180
8	$(8, 2, 2)$	0.9043
9	$(9, 2, 1)$	0.9008
9	$(9, 1, 2)$	0.9012
10	$(10, 1, 1)$	0.8958

The ordering by d_0 is strictly monotonic. At fixed $d_0 = 9$, the two configurations differing in tail allocation (d_1, d_2) agree to $\Delta\text{BPB} = 0.0004$, well within run-to-run variability.

8.4 Auxiliary Observations

Three auxiliary observations emerge from the data, noted here as potential entry points for future structural work.

(Obs. 1) Diminishing returns in BPB improvement per layer. The per-unit-d_0 improvement is approximately 0.008 BPB in the $d_0 \in \{3, \dots, 6\}$ regime and approximately 0.004 BPB in the $d_0 \in \{8, \dots, 10\}$ regime — a two-fold slowdown. The exact asymptotic as $d_0 \to L$ is not characterized.

(Obs. 2) Tail symmetry at $d_0 = 9$. The configurations $(9, 2, 1)$ and $(9, 1, 2)$ differ only in the ordering of tail depths, and yield empirically indistinguishable BPB trajectories ($\Delta = 0.0004$). This is consistent with a structural theorem (proved elsewhere, cf. Paper 1 §5.3) that the tail stages with per-stage heads and uniform aggregation are exchangeable up to a permutation — a symmetry of the training operator under tail permutations.

(Obs. 3) Cross-checkpoint stability. The BPB ordering across d_0 values is stable from step 400 onward, with the gap between configurations modestly widening through training. This indicates that Lemma 8.1 is a robust training-regime phenomenon, not a late-training artifact.

8.5 Position of the Lemma

Remark 8.2 (What the Lemma Does Not Say). Lemma 8.1 is a statement about the BPB metric along a specific one-dimensional slice. It does *not* say:

- That deep Stage 0 is preferable in any deployment. BPB is one property; the slice $\mathcal{S}$ is a one-dimensional cross-section of a multi-property configuration space.
- That $(10, 1, 1)$ is a better configuration than $(4, 4, 4)$. The former has negligible capability-matrix properties (§7.5); the latter lies in the capability region.
- That BP is approached at the $d_0 = 10$ end. It is approached in BPB; other properties diverge (see §8.6).

The Lemma is presented as a demonstration that $\mathcal{C}(L)$ admits structure, not as a recommendation for deployment.

8.6 Relation to Property Inheritance

The one-dimensional family $\mathcal{S}$ traces a curve in the property space of §7. As d_0 increases from 2 to 10:

- **BP-descending property (task BPB)** improves: BPB decreases.
- **Local-ascending properties** (memory locality, pipeline parallelism) modestly improve due to smaller tail stages, but dominated by the growth of Stage 0 itself.
- **Configuration-sensitive properties** (capability matrix) *decline*: the tail stages become too shallow to support specialist capacity (by $d_0 = 10$ the tail stages are single layers); prefix exit points move to trivially late; parallel composition gain diminishes.

The Lemma's monotonicity in BPB is thus accompanied by a non-monotonicity (indeed, decline) in configuration-sensitive properties. A multi-dimensional characterization of this curve is the subject of §10.

8.7 The Root Bottleneck Principle

We now state the structural principle of which Lemma 8.1 is the empirical signature. The principle is information-theoretic and depends only on the chain topology of the forward graph, not on the specific Transformer architecture or dataset used in §§8.2–8.3.

Definition 8.3 (Root representation). For a chain partition $\pi = (d_0, \dots, d_{K-1}) \in \Pi(L)$, the *root representation* of input x under (π, H) is

$$r_\pi(x) = h_{\ell_0(\pi)}, \tag{19}$$

i.e., the hidden state at the terminus of Stage 0.

Assumption 8.4 (Chain forward dependency without bypass). Under the architecture of §§2–4, every stage $k \geq 1$ receives its input solely from the previous stage's terminal hidden state; there is no bypass path from the input embedding or from any earlier hidden state (other than $h_{\ell_{k-1}(\pi)}$) to stage k's first layer.

This is the architectural assumption implicit in the stage-partitioned training operator of §4, and it is the same assumption under which the local-learning endpoint theorem (§6) was stated.

Proposition 8.5 (Root bottleneck). Under Assumption 8.4, for every $k \geq 1$ and every $(\pi, H) \in \mathcal{C}(L)$, there exists a deterministic map $G_{k,\pi,H}$ such that

$$h_{\ell_k(\pi)} = G_{k,\pi,H}\big(r_\pi(x)\big), \tag{20}$$

and consequently the full Log-OP aggregate logit

$$z_{\text{full}}(x; \pi, H) = \sum_{k=0}^{|\pi|-1} z_k(\pi, H) \tag{21}$$

and the induced posterior $p_{\text{full}}(y \mid x; \pi, H)$ are functions of $r_\pi(x)$ alone.

Proof. By Assumption 8.4, stage k's forward pass is determined by its input $h_{\ell_{k-1}(\pi)}$ and its parameters. Inductively, $h_{\ell_k(\pi)}$ is a function of $h_{\ell_0(\pi)} = r_\pi(x)$ alone. The stage logits $z_k(\pi, H)$ depend only on $h_{\ell_k(\pi)}$ (and the head parameters), hence on $r_\pi(x)$. The sum z_{full} and its softmax inherit this dependency. □

Corollary 8.6 (Data processing bound). Under Assumption 8.4, for every $k \geq 0$,

$$I\big(Y; h_{\ell_k(\pi)}\big) \leq I\big(Y; r_\pi(x)\big), \tag{22}$$

with equality when $G_{k,\pi,H}$ is invertible on the support of $r_\pi(x)$.

Proof. Immediate from the Markov chain $Y \to X \to r_\pi(x) \to h_{\ell_k(\pi)}$ and the data processing inequality. □

Principle 8.7 (Root Bottleneck Principle). For a chain-partitioned local-learning model without independent input access to later stages, the first-stage representation $r_\pi(x)$ is an information bottleneck for all later stages. Consequently, the sufficiency of $r_\pi(x)$ for Y bounds the attainable log-likelihood (equivalently, the attainable BPB) of any aggregate posterior in $\mathcal{C}(L)$.

The principle has two immediate structural consequences.

Corollary 8.8 (Stage 0 depth bias). The depth d_0 controls both the representational capacity and the local gradient horizon available to r_π. Any selection rule that prefers configurations with smaller $\mathbb{E}[-\log p_{\text{full}}(Y \mid X)]$ is therefore biased toward increasing d_0.

Corollary 8.9 (Path to the BP endpoint). In the limit $d_0 \to L$, the chain has no internal boundary, $|\pi| \to 1$, and (π, H) approaches the BP endpoint $\hat{0}$. Equivalently, the Stage 0 depth axis is the coordinate along which BPB-driven selection traces a path from any interior configuration to BP.

Remark 8.10 (Stage 0 depth is not merely a hyperparameter). Proposition 8.5 reinterprets Lemma 8.1. Stage 0 depth is not a free hyperparameter to be tuned; it is the coordinate along which BPB pulls the configuration space back toward BP. The monotonicity of Lemma 8.1 is a structural, not empirical, regularity modulo run-to-run noise.

8.8 BPB as a BP-Descending Projection

The root bottleneck principle has a sharper consequence when we consider *optimization* over $\mathcal{C}(L)$.

Proposition 8.11 (BPB collapse). Let $\overline{\text{BPB}} : \mathcal{C}(L) \to \mathbb{R}$ be any utility that is monotone-decreasing in next-token log-likelihood (BPB itself, perplexity, cross-entropy). Let $\phi : \mathcal{C}(L) \to \mathcal{C}(L)$ be any selection operator that moves toward lower $\overline{\text{BPB}}$ with no other regularizer. Then under Assumption 8.4, the fixed points of ϕ lie at the BP endpoint $\hat{0} = ((L), \text{sh})$ (equivalently, at its gauge orbit, §5.3).

Proof sketch. By Proposition 8.5, p_{full} is a function of $r_\pi(x)$. Moving any internal boundary inward (merging a boundary between stages k and $k+1$ into a deeper Stage 0) weakly enlarges d_0 and hence weakly enlarges the representational and gradient horizon of r_π. Equivalently, by Corollary 8.6 the mutual information $I(Y; r_\pi(x))$ is the ceiling on what any aggregate can achieve; relaxing stop-gradients along Stage 0 weakly reduces this ceiling's slack. Hence every refinement-removal step is BPB-non-increasing. Iterating ϕ removes boundaries one by one and terminates at $|\pi| = 1$. □

Corollary 8.12 (BPB is not neutral over $\mathcal{C}(L)$). BPB is a *BP-descending* property in the sense of Definition 7.1. BPB-only utility functionals have their optimum at $\hat{0}$; no interior configuration is BPB-optimal.

Remark 8.13 (The correct framing of dynamic selection). Proposition 8.11 is not a failure of the framework but a prediction of it. The whole point of §7's property decomposition is that different properties descend in different directions of $\mathcal{C}(L)$. BPB descends toward BP; memory locality, parallelism granularity, biological plausibility, and the capability matrix ascend (or peak in the interior). Any practically interesting configuration-selection procedure must therefore combine BPB with at least one non-BP-descending term. We develop this as a multi-objective problem in §10.1, and show in §10.3 that dynamic boundary selection under BPB-only objectives is degenerate in exactly the sense of Proposition 8.11.

Remark 8.14 (What the principle does and does not rule out). Proposition 8.5 depends on Assumption 8.4 — no bypass. If the chain assumption is broken (e.g., later stages receive independent input streams, or the graph fans out from the input into parallel experts), then later stages are no longer functions of $r_\pi(x)$, the ceiling $I(Y; r_\pi(x))$ no longer applies, and BPB ceases to be monotone in d_0. This is the structural content of §10.4–10.5: *to obtain interior configurations competitive on BPB without surrendering the capability matrix, one must expand the architecture beyond chain partitions.*

8.9 The Root Bandwidth Bottleneck

The information-theoretic bottleneck of §§8.7–8.8 is necessary but not sufficient to explain the empirical slope of Lemma 8.1. Jeong (2026c) establishes a complementary, capacity-theoretic mechanism: **answer bandwidth**. A representation $h \in \mathbb{R}^d$ carries bandwidth in proportion to its effective rank

$$r_{\text{eff}}(h) = \exp\Big(-\textstyle\sum_i \hat{\sigma}_i \log \hat{\sigma}_i\Big), \qquad \hat{\sigma}_i = \sigma_i / \textstyle\sum_j \sigma_j, \tag{23}$$

where σ_i are the singular values of the activation matrix. A bounded relay — any operator mapping into a compact range such as $[0,1]^d$ or $[-1,1]^d$ without a multiplicative bypass — compresses r_{eff} and thereby destroys answer bandwidth, independently of gradient flow. The six-experiment arc of Jeong (2026c) establishes this on CIFAR-10 (VGG-style CNN) and WikiText-2 (GPT-2) by showing that sigmoid accuracy and r_{eff} rise with a bandwidth parameter β, that Swish and GELU (whose $x \cdot g(x)$ form retains the multiplicative x factor) are near-β-invariant on CNNs, and that Transformer residual streams *partially* rescue sigmoid while additive skip connections do *not*. Two implications transfer directly.

Observation 8.15 (Boundary operators as bandwidth bottlenecks). A boundary in $\mathcal{C}(L)$ is an abstract object (an insertion of sg). Realized as an *architectural* operator — a projection to logit space, a sigmoid gate, a bounded normalization — it acts both on gradient flow (detachment) and on the forward activation pathway (bandwidth). If the boundary operator is bounded without multiplicative bypass, it compresses r_{eff} of the representation that downstream stages receive, and therefore degrades every later stage's achievable posterior.

Proposition 8.16 (Root bandwidth bottleneck). Under Assumption 8.4 together with a bounded boundary realization (i.e., the stage-0-to-stage-1 transition is a bandwidth-compressing map), the representation $r_\pi(x)$ has strictly lower effective rank than the corresponding BP hidden state at the same layer index. Consequently, the ceiling $I(Y; r_\pi(x))$ of Corollary 8.6 is tighter than the pure-gradient-detachment bound would predict, and Lemma 8.1's monotonicity in d_0 reflects two additive mechanisms: an information-theoretic bottleneck (§8.7) and a bandwidth bottleneck (present proposition).

Proof sketch. Under a bounded relay without multiplicative bypass, Jeong (2026c) shows $r_{\text{eff}}(\text{bounded}(h)) < r_{\text{eff}}(h)$ on typical activation distributions. Iterating through layers and applying the data processing inequality yields $I(Y; \text{bounded-relay}(r_\pi(x)))$ $\leq I(Y; r_\pi(x))$. Since the subsequent chain depends only on this strictly-smaller quantity, the bound of Corollary 8.6 is tightened. □

Corollary 8.17 (Two-component reading of Stage 0 depth dominance). The monotonic decrease of BPB in d_0 (Lemma 8.1) decomposes as:

$$\Delta\text{BPB}(d_0 \to d_0+1) = \underbrace{\Delta\text{BPB}_{\text{info}}}_{\text{info-theoretic ceiling}} + \underbrace{\Delta\text{BPB}_{\text{bw}}}_{\text{bandwidth preservation}} + o(1), \tag{24}$$

where the first term is the root-bottleneck ceiling slack (§8.7) and the second is the answer-bandwidth slack (present §8.9). The two-fold slowdown observed in §8.4 Obs. 1 (per-layer improvement falling from ≈ 0.008 to ≈ 0.004 BPB as d_0 grows from $\{3, \ldots, 6\}$ to $\{8, \ldots, 10\}$) is consistent with the bandwidth term saturating earlier than the information-ceiling term.

Principle 8.18 (Detachment vs. information deletion). A stage boundary is defined operationally as a point at which gradients are zeroed. It is *not* defined as a point at which information or bandwidth is destroyed. When a boundary is implemented by a bounded operator without multiplicative bypass, the two become accidentally conflated, and the framework inherits a bandwidth penalty it did not intend. The correct summary is:

> **Detachment is a gradient operation, not an information-deletion operation.**

This principle motivates both the detached input bus of §10.5.1 (restore information flow while keeping sg) and the multiplicative boundary gate of §10.5.5 (preserve bandwidth while keeping detachment).

Remark 8.19 (Scope of the bandwidth lens). Answer bandwidth is an *intra-activation* property of the operator applied at a boundary, not of the partition π itself. Two different architectural realizations of the same (π, H) — one with an identity relay, one with a sigmoid relay — yield different bandwidth profiles and, empirically, different BPB trajectories. This is a further reason to make R a first-class coordinate of $\mathcal{C}^+(L)$ (§10.4).

Remark 8.20 (Question sequencing). Jeong (2026b) conjectured a compositional ordering principle for activation types. Jeong (2026c) refuted this compositional prediction: the dominant variable is bandwidth, not question semantics. We incorporate this as follows: *the activation-as-question vocabulary remains descriptively useful, but stage-compositional performance is governed by answer bandwidth.* This framing absorbs both prior results without contradiction.

---

9. Soft Boundaries and the Relaxed Configuration Space

The configuration lattice $\mathcal{C}(L)$ is discrete, with $|\mathcal{C}(L)| = 2^L$. For dynamic (training-time, gradient-based) boundary selection, and for making Proposition 8.11 quantitative on a differentiable objective, a continuous relaxation is required. This section constructs it.

9.1 Boundary Vector Reformulation

Definition 9.1 (Boundary vector). For a composition $\pi \in \Pi(L)$, the *boundary indicator* $b(\pi) \in \{0,1\}^{L-1}$ is

$$b(\pi)_i = \mathbb{1}\big[i \in \{\ell_0(\pi), \dots, \ell_{|\pi|-2}(\pi)\}\big], \quad 1 \le i \le L-1. \tag{25}$$

By Proposition 2.2, the map $\pi \mapsto b(\pi)$ is a bijection $\Pi(L) \leftrightarrow \{0,1\}^{L-1}$. BP corresponds to $b \equiv 0$; fully local corresponds to $b \equiv 1$. The refinement order $\preceq$ on $\Pi(L)$ is the componentwise order on $\{0,1\}^{L-1}$.

This reformulation is merely notational at the level of $\Pi(L)$ but clarifies the continuous relaxation that follows.

9.2 Soft Detachment

Definition 9.2 (Layer-wise gradient transmissivity). For each internal layer index $i \in \{1, \dots, L-1\}$ we assign a scalar $\lambda_i \in [0,1]$, the *gradient transmissivity* at layer i. The *boundary relay* is the operator

$$D_{\lambda_i}(h) = \mathrm{sg}(h) + \lambda_i \cdot \big(h - \mathrm{sg}(h)\big). \tag{26}$$

By the definition of sg (Definition 4.4), D_{λ_i} satisfies:

- *Forward pass:* $D_{\lambda_i}(h) = h$ for all λ_i (the sg term and its complement sum to h as values).
- *Backward pass:* $\partial D_{\lambda_i}/\partial h = \lambda_i \cdot I$.

Hence:

- $\lambda_i = 1$: gradient passes through unchanged. No boundary.
- $\lambda_i = 0$: gradient is fully detached. Hard boundary (identical to sg).
- $\lambda_i \in (0,1)$: gradient is attenuated. Soft boundary.

9.3 The Relaxed Configuration Space

Definition 9.3 (Relaxed configuration space). The *relaxed configuration space* at depth L is

$$\overline{\mathcal{C}}(L) = [0,1]^{L-1} \times \mathbf{H}. \tag{27}$$

An element is a pair (λ, H) with $\lambda = (\lambda_1, \dots, \lambda_{L-1})$.

Proposition 9.4 (Embedding of hard configurations). The map

$$\iota : \mathcal{C}(L) \to \overline{\mathcal{C}}(L), \quad (\pi, H) \mapsto \big(\mathbf{1} - b(\pi), H\big) \tag{28}$$

is a bijection onto the vertex set $\{0,1\}^{L-1} \times \mathbf{H}$ of $\overline{\mathcal{C}}(L)$. Under ι, the training operator of Definition 4.5 extends consistently to $\overline{\mathcal{C}}(L)$ by replacing each internal sg operation at layer i with the boundary relay D_{λ_i}.

Proof sketch. At a hard vertex, $\lambda_i \in \{0,1\}$. When $\lambda_i = 0$, $D_{\lambda_i} = \mathrm{sg}$; when $\lambda_i = 1$, $D_{\lambda_i} = \mathrm{id}$. The hard vertex where $\lambda_i = 1 - b(\pi)_i$ for all i therefore reproduces exactly the configuration (π, H). □

Definition 9.5 (Soft stage-partitioned loss). For $(\lambda, H) \in \overline{\mathcal{C}}(L)$, the *soft stage-partitioned loss* attaches a per-layer loss with weight $\tau_i = 1 - \lambda_i$ (interior layers) and $\tau_L = 1$ (terminal layer), representing the intensity with which layer i acts as a stage terminus:

$$\mathcal{L}^{\lambda,H}(\theta; x, y) = \tau_L \cdot \mathrm{CE}\big(z_L^{(H)}, y\big) + \sum_{i=1}^{L-1} \tau_i \cdot \mathrm{CE}\big(z_i^{(H)}, y\big), \tag{29}$$

with $z_i^{(\mathrm{sh})} = W_{\mathrm{hd}} h_i$ and $z_i^{(\mathrm{ps})} = (W_{\mathrm{hd}} + W_{\mathrm{hd},i}) h_i$.

Remark 9.6 (Hard-limit compatibility). At a hard vertex $\lambda_i \in \{0,1\}$, $\tau_i \in \{0,1\}$ and a nonzero τ_i occurs exactly at stage termini of the corresponding composition π. The soft loss then coincides with the hard loss of Definition 4.2 up to the gauge equivalence of §5.3 (a stage with $\tau = 0$ at its internal layers contributes no loss and is indistinguishable from a single-layer of the same stage). Hence $\overline{\mathcal{C}}(L)$ is a genuine extension, not a perturbation.

Remark 9.7 (Position of §9 relative to the framework). The relaxation $\overline{\mathcal{C}}(L)$ is introduced for a single purpose: to make the BPB collapse proposition (§8.8) a statement about a *differentiable* objective on a *continuous* domain, and to make the dynamic boundary selection of §10 differentiable. It is not a physically distinct family of training algorithms; at any hard vertex it reduces to $\mathcal{C}(L)$.

10. Multi-Objective Selection, Dynamic Boundaries, and Architecture-Expanded Configuration Spaces

The framework of §§2–9 is sufficient to pose the configuration-selection problem, but — by Proposition 8.11 — insufficient to solve it non-trivially. This section develops the three extensions that are necessary: a multi-objective utility (§10.1); a dynamic procedure for searching over $\overline{\mathcal{C}}(L)$ with at least one non-BP-descending term (§§10.2–10.3); and an architecture-expanded configuration space $\mathcal{C}^{+}(L)$ (§§10.4–10.5) that breaks the root bottleneck's chain assumption. §§10.6–10.7 develop the concrete research program and experimental design.

10.1 Multi-Objective Configuration Utility

By Proposition 8.11, any utility of the form $U = -\mathrm{BPB}$ is maximized at the BP endpoint and cannot select an interior configuration. The minimal correct formulation adds locality and capability terms.

Definition 10.1 (Multi-objective configuration utility). A *multi-objective configuration utility* on $\overline{\mathcal{C}}(L)$ is a functional

$$J(\theta, \lambda, H) = \mathrm{BPB}(\theta, \lambda, H) + \beta \cdot C_{\mathrm{global}}(\lambda) + \gamma \cdot C_{\mathrm{head}}(\lambda, H) - \eta \cdot U_{\mathrm{cap}}(\lambda, H) + \rho \cdot C_{\mathrm{bw}}(\theta, \lambda) + \kappa \cdot C_{\mathrm{cal}}(\theta, \lambda, H), \quad (30)$$

where:

- $C_{\mathrm{global}}(\lambda) = \sum_{i=1}^{L-1} \lambda_i \cdot c_i$ is the *gradient coupling cost*, weighted by per-layer costs $c_i \geq 0$;
- $C_{\mathrm{head}}(\lambda, H)$ is the *head and projection parameter cost* under (λ, H);
- $U_{\mathrm{cap}}(\lambda, H)$ is a *capability utility* combining prefix-exit utility, WAND pruning gain, specialist capacity, parallel composition gain;
- $C_{\mathrm{bw}}(\theta, \lambda) = \sum_{i=1}^{L-1} (1 - \lambda_i) \cdot \max\big(0, r_{\min} - r_{\mathrm{eff}}(h_i)\big)$ is the *bandwidth-bottleneck cost* (§8.9): placing a hard boundary at a low-bandwidth layer is penalized in proportion to how far below the bandwidth floor $r_{\min}$ the representation sits;
- $C_{\mathrm{cal}}(\theta, \lambda, H)$ is the *calibration cost* (§7.6): the expected calibration error of each stage-local posterior $p_k(y \mid x)$ under its own head, measured as e.g. reliability-diagram ECE, averaged over active stages.

Proposition 10.2 (Interior optimum of J). For $\beta, \eta > 0$ and non-trivial c_i and U_{cap}, the minimum of J over $\overline{\mathcal{C}}(L)$ lies in the interior, not at $\lambda \equiv 1$ (BP) nor at $\lambda \equiv 0$ (fully local). When $\rho > 0$, the bandwidth term additionally penalizes interior configurations whose boundaries fall at low-bandwidth layers, pulling the optimum toward boundaries that sit at genuinely rank-sufficient representations.

Proof sketch. At $\lambda \equiv 1$, C_{global} is maximal and $U_{\mathrm{cap}} = 0$, so J is penalized. At $\lambda \equiv 0$, BPB is maximal (all stages are single layers, representing the finest local learning with capacity constraints) and U_{cap} is again zero or small (P9 of §7 requires non-trivial stages). Under any smooth continuation, the infimum is attained at an interior point. The bandwidth term C_{bw} is identically zero at $\lambda \equiv 1$ (no active boundaries), so it does not change the minimum's avoidance of the BP endpoint; it selects among interior configurations. □

Remark 10.3 (Pareto reading). Equivalently, one can view the problem as Pareto optimization over the multi-objective vector $(-\mathrm{BPB}, -C_{\mathrm{global}}, -C_{\mathrm{head}}, U_{\mathrm{cap}}, -C_{\mathrm{bw}}, -C_{\mathrm{cal}})$ and parameterize $(\beta, \gamma, \eta, \rho, \kappa)$ as the utility weights of a deployment context. Different deployments will select different Pareto-optimal points. Datacenter pretraining with abundant compute and no capability requirements reduces $(\beta, \gamma, \eta, \rho, \kappa) \to 0$ and recovers BP. On-device deployment with memory, energy, and modularity constraints sets β, η, ρ large and selects interior configurations with bandwidth-sufficient boundaries and well-calibrated stage heads.

Remark 10.3a (Four boundary-selection criteria). The utility J embeds four distinct criteria that boundary selection must satisfy jointly: (i) the local predictor must be accurate (BPB term); (ii) the hidden state at the boundary must carry sufficient bandwidth (C_{bw}); (iii) the stage-local posterior must be well-calibrated (C_{cal}); and (iv) the downstream structure must retain residual-correction room and capability-matrix properties (U_{cap}). A boundary that minimizes BPB alone — the Proposition 8.11 degenerate case — fails at least (ii)–(iv).

10.2 Training-Time Dynamic Boundary Selection

The relaxation of §9 makes λ a learnable vector. Three concrete procedures fall out.

10.2.1 Soft Boundary Learning

Parameterize $\lambda_i = \sigma(a_i / T)$ with learnable logits a_i and a temperature T annealed from $T_0 \gg 1$ toward $T_\infty \ll 1$ (hardening schedule). The joint objective is

$$\min_{\theta, a} \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\mathcal{L}^{\lambda(a), H}(\theta; x, y)\right] + \beta \sum_i \lambda_i(a) + \gamma\, R_{\mathrm{stage}}(\lambda(a)), \quad (31)$$

with stage-length regularizer

$$R_{\text{stage}}(\lambda) = \sum_k \max\Big(0, d_{\min} - \hat{d}_k(\lambda)\Big)^2, \tag{32}$$

where $\hat{d}_k(\lambda)$ is the expected stage-k depth under the Bernoulli distribution with parameters $(1 - \lambda_i)$. Under temperature annealing, $\lambda_i \to \{0, 1\}$ and the configuration hardens to a vertex of $\mathcal{C}(L)$.

Remark 10.4. This is the most powerful procedure in principle, because λ genuinely alters the gradient topology during training. It requires per-layer heads at all candidate boundary layers, which costs parameters but is $O(L)$ and therefore asymptotically negligible compared to the $O(L \cdot d^2)$ hidden-parameter cost.

10.2.2 Periodic Boundary Surgery

An alternative maintains a hard $\pi_t \in \Pi(L)$ throughout training but periodically re-evaluates it.

Algorithm (Periodic boundary surgery).

1. Initialize $\pi_0 \in \Pi(L)$; train for N steps under the training operator $T(\pi_0, H)$.
2. At step t, measure *boundary salience* on a validation batch (three variants below).
3. Generate split/merge/move candidate partitions $\{\pi_t^{(1)}, \pi_t^{(2)}, \ldots\}$.
4. Select π_{t+1} by a greedy rule on J or by stochastic acceptance.
5. Carry over hidden-parameter state; recalibrate (or reinitialize and warm up) the per-stage heads.

Three boundary-salience measures are natural.

Merge pressure at boundary i. Let $\text{merge}_i(\pi)$ denote π with the i-th internal boundary removed. Then

$$M_i = \big[\text{BPB}\big(\text{merge}_i(\pi)\big) - \text{BPB}(\pi)\big] - \beta\big[C_{\text{global}}\big(\text{merge}_i(\pi)\big) - C_{\text{global}}(\pi)\big]; \tag{33}$$

a negative M_i means the merge is preferred. One does not literally retrain; instead, M_i is estimated by a brief forward-pass evaluation under relaxed λ_i.

Split pressure at layer i inside a stage. Let $\text{split}_i(\pi)$ denote π with a new boundary introduced at layer i.

$$S_i = \eta\big[U_{\text{cap}}\big(\text{split}_i(\pi)\big) - U_{\text{cap}}(\pi)\big] - \big[\text{BPB}\big(\text{split}_i(\pi)\big) - \text{BPB}(\pi)\big] - \gamma \Delta C_{\text{head}}(i). \tag{34}$$

A positive S_i means the split is preferred.

Tail usefulness. For stage k, with cumulative partial aggregate $A_{k-1} = \sum_{j<k} z_j$,

$$U_k = \text{CE}(A_{k-1}, y) - \text{CE}(A_{k-1} + z_k, y). \tag{35}$$

A small U_k indicates that stage k is a near-redundant expert; the boundary at ℓ_{k-1} is then a candidate for merge or for an objective change (§10.5.4).

10.2.3 Constrained Dynamic Programming

When a segment-cost $Q(a, b)$ can be estimated for each candidate interval $[a + 1, b]$ (e.g., by a shallow probe trained for K steps), the optimal K-stage partition satisfies

$$\text{DP}[m, b] = \min_{a<b}\big\{\text{DP}[m - 1, a] + Q(a, b) + R(a, b)\big\}, \tag{36}$$

with $\text{DP}[1, b] = Q(0, b) + R(0, b)$ and $R(a, b)$ a stage-length / capability penalty. This gives a principled (though expensive) answer for *fixed* K, and is useful primarily as a benchmark against which §10.2.1–10.2.2 are compared.

10.3 Post-Hoc Boundary Selection

Post-hoc boundary selection must be handled carefully, because a boundary is not merely an inference-time choice: it was a gradient-topology choice during training. Two regimes are structurally distinct.

10.3.1 True Post-Hoc: Inference-Path Selection on a Frozen Model

Given a model trained under some (π_0, H_0), one can at inference time:

- select a prefix exit at layer $\ell_k(\pi_0)$ for any k;
- apply WAND-style aggregation pruning over stages;
- select a branch composition $\{[a_j, b_j]\}$ of trained substructures;
- route to a trained specialist stage.

These are the capability-matrix operations of Paper 1 (F2, F3, F4, F6). They are not post-hoc *configurations* in the sense of $\mathcal{C}(L)$; they are inference-time path choices over a frozen (π_0, H_0) representation. The existing composition of π_0 is fixed; no new training-time boundaries are introduced.

10.3.2 Supernet-Style Post-Hoc: Hardening a Boundary-Superposition Model

Genuine post-training boundary selection requires *training-time preparation*: one trains a boundary-superposition model (§10.2.1 with soft λ) and then hardens λ to a vertex after training. The resulting configuration is a new (π, H) whose training dynamics were precisely those of the soft-λ intermediate; no retraining is needed, but the training-time gradient topology was committed in advance.

Remark 10.5 (Post-hoc boundary selection is path selection, not topology selection). Post-hoc boundary selection without boundary-superposition training can select inference paths, but cannot retroactively change the gradient topology that produced the representations. This distinction is often elided in practice and is worth stating explicitly.

10.4 The Architecture-Expanded Configuration Space $\mathcal{C}^+(L)$

Proposition 8.11 (BPB collapse) depends on four assumptions, and Proposition 8.16 (root bandwidth bottleneck) adds a fifth:

1. the forward graph is a chain;
2. later stages depend on Stage 0's hidden state only;
3. every stage predicts the same next-token distribution directly;
4. the selection metric is BPB-dominated;
5. the boundary operator is bounded without multiplicative bypass (i.e., a bandwidth-restricting relay).

§10.1 addressed (4). To weaken (1)–(3) and (5), we extend $\mathcal{C}(L)$ as follows.

Definition 10.6 (Architecture-expanded configuration space). The *architecture-expanded configuration space* at depth L is

$$\mathcal{C}^+(L) = \big\{(G, S, H, O, R)\big\}, \tag{37}$$

with:

- G: a DAG on $L+1$ nodes $\{0, 1, \dots, L\}$, representing the stage graph;
- $S : \{1, \dots, L\} \to \mathcal{P}(\{0, \dots, L-1\})$: the *source map*, specifying for each layer which earlier hidden states it receives as input (with stop-gradients on non-predecessor sources, see §10.5.1);
- H: the head structure;
- $O \in \{\text{full\_CE}, \text{residual\_CE}, \dots\}$: the *objective family*;
- $R \in \mathcal{R}$: the *boundary relay structure*, specifying at each internal boundary which operator realizes the detachment (identity sg, multiplicative gate $h \odot \sigma(g(h))$, residual-stream pre-norm relay, low-rank relay, etc.). The space $\mathcal{R}$ is architecture-dependent; we treat it as a finite catalogue whose entries are detailed in §10.5.

Proposition 10.7. The chain-partition configuration space $\mathcal{C}(L)$ is the sub-case

$$\mathcal{C}(L) = \big\{(G_{\text{chain}}, S_{\text{prev}}, H, O_{\text{full\_CE}}, R_{\text{sg}})\big\} \subset \mathcal{C}^+(L), \tag{38}$$

where G_{chain} is the path graph $0 \to 1 \to \cdots \to L$, $S_{\text{prev}}(i) = \{i-1\}$, $O_{\text{full\_CE}}$ is the per-stage cross-entropy loss of Definition 4.2, and R_{sg} is the identity stop-gradient relay of Definition 4.4.

Proof. Immediate by construction. Under chain G, prev-source S, full-CE O, and identity-sg relay R, the forward pass and loss of Definition 4.2 are recovered verbatim. □

Remark 10.8 (Why expand?). The root bottleneck principle (Proposition 8.5) and its bandwidth sharpening (Proposition 8.16) are *statements about* $\mathcal{C}(L)$. The first depends on Assumption 8.4, which is precisely the restriction that $G = G_{\text{chain}}$ and $S = S_{\text{prev}}$. The second depends on $R = R_{\text{sg}}$ composed with a bounded architectural operator — a *bandwidth-restricting relay*. In $\mathcal{C}^+(L)$, none of these three restrictions holds in general; later stages can receive independent input streams, stages can be organized as forests or DAGs, and relays can be designed to preserve bandwidth. In these cases Proposition 8.5 ceases to apply (information), Proposition 8.16 ceases to apply (bandwidth), and BPB is no longer a BP-descending projection along a single "Stage 0 depth" coordinate.

10.5 Five Structural Mitigations of the Root Bottleneck

Within $\mathcal{C}^+(L)$ there are five canonical extensions that weaken one or more of Proposition 8.11's assumptions, or Proposition 8.16's bandwidth bound.

10.5.1 Detached Input Bus

Definition 10.9. A *detached input bus* is a modification of S in which each stage k receives, in addition to $h_{\ell_{k-1}(\pi)}$, one or more early sources $\{h_{s_1}, \dots, h_{s_m}\} \cup \{e_0\}$, each wrapped in $\text{sg}(\cdot)$:

$$\tilde{h}^{(k)}_{\ell_{k-1}(\pi)} = \text{sg}(h_{\ell_{k-1}(\pi)}) \oplus \text{sg}(e_0) \oplus \text{sg}(h_{s_1}) \oplus \cdots. \tag{39}$$

The forward pass of stage k begins from this concatenated (or summed, or gated) input. Gradients through the early-source paths are zero.

Proposition 10.10 (Root bottleneck weakening under detached bus). Under a detached input bus carrying e_0 to all later stages, the conclusion of Proposition 8.5 is replaced by $h_{\ell_k(\pi)} = G_k(r_\pi(x), e_0)$. In particular, if e_0 carries information about Y that $r_\pi(x)$ has discarded (e.g., rare-token embedding detail), the data-processing bound of Corollary 8.6 is no longer a ceiling on p_{full}.

Remark 10.10a (Operational slogan). The detached input bus is the architectural instantiation of Principle 8.18: *detachment is a gradient operation, not an information-deletion operation.* Gradient isolation (sg) is preserved; information flow via the bus is restored. Paper 1's x_0 blending in the nanochat family is a weak form of this; the empirical persistence of Stage 0 dominance in §§8.1–8.6 indicates that raw embedding relay alone is insufficient in low-budget regimes. A richer bus — low-rank lexical/positional relays, learned shallow representations, or the pre-norm residual stream that Jeong (2026c) shows partially rescues sigmoid in Transformers, all under sg — is the natural strengthening.

10.5.2 Stage Forest / Parallel Roots

Definition 10.11. A *stage forest* is a configuration $(G, S, H, O) \in \mathcal{C}^+(L)$ in which G is a forest with multiple roots, each rooted at the input node 0. Each branch B_b has its own layer budget, its own head, and its own local CE loss; all branches' logits are combined by Log-OP:

$$z_{\text{full}}(x) = \sum_{b=1}^{B} z_b(x). \tag{40}$$

Formally, G fans out from node 0 into B disjoint chains $\{0 \to B_b^{(1)} \to \cdots \to B_b^{(L_b)}\}$ with $\sum_b L_b = L$.

Proposition 10.12. A stage forest has no single root bottleneck; the Proposition 8.5 map $h_k = G_k(r_\pi(x))$ does not hold because there is no unique $r_\pi(x)$. Instead, each branch has its own root representation, and the aggregate p_{full} combines B independent experts.

The framework of Paper 1 §5.3 (parallel composition of $[1..4]$ and $[1..5]$) is a partial instance with branches sharing a prefix. A pure forest shares no interior nodes; its branches are architecturally decoupled except through the aggregate loss. The total layer budget L is distributed across branches rather than accumulated sequentially.

10.5.3 Overlapping Interval Experts

Definition 10.13. An *interval-cover* configuration replaces the partition of $\{1, \dots, L\}$ by an interval cover $\mathcal{I} = \{[a_k, b_k]\}_{k=1}^{K}$, not necessarily disjoint. Each expert k produces a logit

$$z_k = W_k \cdot h_{[a_k, b_k]}, \tag{41}$$

where $h_{[a_k,b_k]}$ is some summary (last hidden state, pooled, or re-computed from h_{a_k-1} through a dedicated sub-chain) of layers a_k through b_k.

When $\mathcal{I}$ is a partition, this recovers $\mathcal{C}(L)$. When intervals overlap, a single layer may participate in multiple experts, and experts may begin at different depths — including at layer 0, which escapes the root bottleneck.

10.5.4 Residual (Boosted) Log-OP Objective

The objective family O in $\mathcal{C}^{+}(L)$ is the most algorithmically significant axis. In the chain $\mathcal{C}(L)$, $O = O_{\text{full\_CE}}$: every stage predicts y directly. An alternative is:

Definition 10.14 (Residual Log-OP objective). Let $A_{k-1}(x) = \sum_{j<k} \mathrm{sg}(z_j(x))$ be the stop-gradient aggregate of earlier stages' logits. The *residual Log-OP loss* is

$$\mathcal{L}^{\text{boost}}(\theta; x, y) = \sum_{k=0}^{|\pi|-1} \mathrm{CE}\Big(A_{k-1}(x) + z_k(x), y\Big), \tag{42}$$

with $A_{-1} = 0$.

Proposition 10.15 (Residual locality). The residual Log-OP loss retains stage-local gradient support: for $j < k$, $\partial \mathcal{L}_k^{\text{boost}} / \partial \theta_j = 0$ because A_{k-1} is stop-gradient.

Proof. $\mathcal{L}_k^{\text{boost}}$'s only gradient-transmitting dependency on earlier stages is through A_{k-1}, which carries no gradient by construction. Hence the gradient on θ_j vanishes for $j < k$, matching the locality of Proposition 4.7. □

Remark 10.16 (Expert role change). Under $\mathcal{L}^{\text{full\_CE}}$, every stage is trained as an independent predictor of y. This is redundant once Stage 0 is representationally capable — which is the root bottleneck story. Under $\mathcal{L}^{\text{boost}}$, stage k is trained as a *residual corrector* of the prior aggregate: it learns what the prior stages missed. This is the classical boosting / additive-model decomposition, applied to logit-space expert aggregation. It is a direct training-time counterpart to Paper 1's inference-time Log-OP / PoE interpretation, and is the minimal change to the objective that converts redundant duplication into complementary specialization.

Structural prediction. Under $\mathcal{L}^{\text{boost}}$, the Stage 0 depth monotonicity of Lemma 8.1 should weaken, tail utility U_k (§10.2.2) should increase for later stages, and the tail allocation symmetry at $d_0 = 9$ (the $(9, 2, 1)$ vs $(9, 1, 2)$ indistinguishability of §8.4) should break if tail stages become genuinely non-fungible residual correctors.

Remark 10.16a (Bayesian reading). The residual Log-OP objective is the training-time counterpart of §7.6's Bayesian evidence interpretation. Each stage now trains toward *complementary evidence* rather than *duplicated full posterior*, which is the natural objective under Log-OP / PoE aggregation: experts that push the current aggregate toward the correct target, not experts that each independently predict it. Paper 1's inference-time parallel composition gain (F4) is the test-time signature of this mechanism; $\mathcal{L}^{\text{boost}}$ installs it at training time.

10.5.5 Multiplicative (Bandwidth-Preserving) Boundary Gates

The four mitigations above break Proposition 8.11's assumptions (chain, prev-source, full-CE). The fifth breaks the implicit bandwidth assumption of Proposition 8.16 — that boundaries are realized by bounded operators without multiplicative bypass.

Definition 10.17 (Multiplicative boundary gate). A *multiplicative boundary gate* at layer i is a boundary relay of the form

$$\tilde{h}_i = h_i \odot \sigma\Big(g_i(h_i)\Big) \quad \text{or} \quad \tilde{h}_i = h_i + r_i \odot \sigma\Big(g_i(h_i)\Big), \tag{43}$$

where g_i is a learned map and r_i a (possibly low-rank) learned relay. Crucially, the multiplicative factor h_i (or r_i) is *unbounded*, so the effective rank of $\tilde{h}_i$ is not compressed by the gate's bounded σ component.

Proposition 10.18 (Bandwidth preservation under multiplicative gate). For a multiplicative gate $\tilde{h}_i = h_i \odot \sigma(g_i(h_i))$, the effective rank $r_{\text{eff}}(\tilde{h}_i)$ is bounded below by a constant fraction of $r_{\text{eff}}(h_i)$ on typical activation distributions, under the assumptions of Jeong (2026c) Experiments 2–3 (CIFAR-10 Swish/GELU). A purely bounded gate $\tilde{h}_i = \sigma(g_i(h_i))$ does not admit such a lower bound.

Proof reference. Jeong (2026c) establishes the bandwidth preservation of the $x \cdot g(x)$ family across 9 configurations on CIFAR-10 (max $|\Delta\text{accuracy}| = 0.44\%$ between Swish and GELU at matched temperature), with Pearson $r = 0.94$ between accuracy and r_{eff}. Bounded $g(x)$ alone admits no such invariance (sigmoid temperature-sensitivity of $\approx 1.23\%$ across $\beta \in [0.25, 4]$). □

Remark 10.19 (Multiplicative gate vs. soft λ). Definition 9.2's boundary relay $D_{\lambda_i}(h) = \mathrm{sg}(h) + \lambda_i(h - \mathrm{sg}(h))$ acts only on gradient; it has no effect on the forward pathway. The multiplicative gate acts only on the forward pathway; it need not affect gradients (it is differentiable end-to-end). The two are orthogonal. In the relaxed space $\overline{\mathcal{C}}(L) \times \mathcal{R}$, one picks λ_i to control gradient topology *and* an element of $\mathcal{R}$ to control forward-pathway bandwidth. The identity choice $R = R_{\text{sg}}$ corresponds to "detach only gradient, do nothing to the activation", which is what the original $\mathcal{C}(L)$ assumes.

Caveat 10.20 (Transformer residual stream). Jeong (2026c) Experiment 6 establishes that the Transformer's pre-norm residual stream provides *partial* bandwidth compensation that CNN additive skips do not. This means that Transformer-based instantiations of $\mathcal{C}(L)$ — including Paper 1 and §§8.1–8.6 — already receive some bandwidth protection from the residual stream, and the marginal benefit of an explicit multiplicative boundary gate may be smaller in Transformer settings than in CNN settings. This is a scope qualifier on the expected effect size of Definition 10.17 in the experiments of §10.7.

10.6 Pareto Frontier Mapping: The Empirical Program

The empirical characterization of $\mathcal{C}^{+}(L)$ is a multi-level program. We preserve the level structure of the original draft but recontextualize each level relative to §§10.1–10.5.

Level 0: Empirical regularity along a one-parameter BPB slice (§§8.1–8.6).
One dimension of one slice of $\mathcal{C}(L)$, by one metric. Completed in this paper.

Level 1: Full property profile along the Level 0 slice.
Along the same slice $\mathcal{S}$, measure not just BPB but activation memory, pipeline-parallelism granularity, capability-matrix properties (Stage 1 exit compute, prefix-pruning accuracy at fixed compute budget, parallel composition logit gain), specialist capacity (dual-head SFT convergence rate, base-preservation fidelity), and inference economics. This makes §7 concrete.

Level 2: Two-dimensional mapping in $\mathcal{C}(L)$.
Extend to (K, d_0) jointly at fixed H. Yields the first genuine Pareto frontier in the framework.

Level 3: Full capability-region mapping in $\mathcal{C}(L)$.
Within $\mathcal{C}_{\text{cap}}(L)$ (§7.5), characterize the Pareto frontier.

Level 4: Architecture-expanded mapping in $\mathcal{C}^{+}(L)$.
Compare the chain sub-case against detached-input-bus, stage-forest, overlapping-interval, residual-Log-OP, and multiplicative-gate extensions at matched layer budget. The first four extensions correspond to breaking one of Proposition 8.11's four structural assumptions (chain, prev-source, full-CE, BPB-only); the fifth breaks the implicit bandwidth assumption of Proposition 8.16.

Level 5: Scale dependence across L and parameter count.
All of the above at multiple model scales ($L = 12, 24, 48, 96$; parameters 300M, 1.3B, 7B, 70B) to characterize how the Pareto frontier scales.

10.7 Experimental Program: Highest-Information Experiments First

At the 286M / $L = 12$ scale used in §8, four experiments discriminate among the structural mitigations at low cost.

Experiment A: Residual Log-OP objective.
Repeat the $K = 3$, $H = \text{ps}$ sweep of §8.2 with $\mathcal{L}^{\text{full\_CE}}$ replaced by $\mathcal{L}^{\text{boost}}$ (§10.5.4). Primary measurements: (i) does the Stage 0 depth monotonicity of Lemma 8.1 weaken or invert? (ii) does tail utility U_k increase for $k > 0$? (iii) does the tail-allocation symmetry $(9, 2, 1) \sim (9, 1, 2)$ break? Implementation cost is minimal: a single line change in the loss. A positive result falsifies the BPB-axis interpretation of the root bottleneck and shows that objective-family changes alone suffice to redistribute information across stages.

Experiment B: Detached input bus.
Augment each stage $k \geq 1$'s input with $\text{sg}(e_0)$, or with a learned low-rank shallow representation $\text{sg}(r_0)$ (§10.5.1). Measurements: (i) BPB improvement at shallow-Stage-0 configurations (e.g., $(4, 4, 4)$); (ii) flattening of the Stage 0 depth slope; (iii) increase in later-stage marginal gain U_k. A positive result quantifies how much of Stage 0 dominance is attributable to the chain-bottleneck assumption (Proposition 8.5 Assumption 8.4) versus to other factors.

Experiment C: Stage forest vs sequential.
Matched layer budget: sequential $(4, 4, 4)$ vs. forest $\{[1..4], [1..4], [1..4]\}$ from x. Measurements: BPB, branch diversity, WAND pruning quality, specialist attachment performance (Paper 1 §6.6). This is the most direct test of whether Stage 0 dominance is a chain-dependency artifact.

Experiment D: Soft boundary learning under two objectives.
Train under soft λ (§10.2.1) with expected boundary count fixed at $K - 1 = 2$ via $\sum_i (1 - \lambda_i) = K - 1$ as a Lagrangian constraint. Run twice: once under BPB-only objective (predicted by Proposition 10.2 to harden at large d_0) and once under J including U_{cap} (predicted to harden at an interior partition). The comparison is the minimal direct test of Proposition 8.11: BPB-only collapse vs. capability-aware interior preservation.

Experiment E: Bandwidth diagnostic and multiplicative gate.
Two sub-experiments testing the bandwidth lens of §8.9. **E1 (diagnostic):** for the §8.2 configurations, measure $r_{\text{eff}}(h_{\ell_0(\pi)})$ at each Stage 0 terminus and correlate with final BPB across the sweep. A strong positive correlation supports Corollary 8.17's two-component reading; weak or null correlation would suggest the root bottleneck is primarily information-theoretic rather than bandwidth-theoretic in the Transformer regime (consistent with Caveat 10.20 on residual-stream rescue). **E2 (intervention):** replace the boundary operator at ℓ_0 with the multiplicative gate of Definition 10.17 on a shallow-Stage-0 configuration such as $(4, 4, 4)$, and measure ΔBPB vs. the identity relay baseline. A negative ΔBPB quantifies the bandwidth component of Stage 0 dominance.

Ordering. Experiment A (residual Log-OP) has the highest information per unit of engineering effort (a single loss-term change) and should be run first. Experiment E1 (bandwidth diagnostic) has the second-highest — a single-pass measurement with no training modification — and is a prerequisite to interpret Experiments B and E2. Experiments B and C require modest architectural work and test different assumptions of Proposition 8.5. Experiment D requires the most infrastructure but is the most structurally informative: it is the direct empirical probe of the BPB collapse proposition.

10.8 Deliverables

Upon completion of Levels 1–5 and Experiments A–E, the research program would deliver:

- A function $U_{\text{profile}} : \mathcal{C}^+(L) \to \mathbb{R}^P$ mapping each configuration to its P-dimensional property profile.
- A catalog of Pareto-optimal configurations under various utility weightings.
- Scaling laws describing how the Pareto frontier evolves with L.
- An empirical characterization of which architectural assumptions (chain, prev-source, full-CE, identity-relay) each Paper 1-scale property depends on.
- A principled configuration-selection procedure for new deployment contexts.

This agenda is stated; its execution is future work.

---

11. Relation to Prior Work: Paper 1 as a Special Case

11.1 The Embedding

The prior work of Jeong (2026d), *Product of Experts as Scalable Local Learning* (referred to as Paper 1), studied a specific configuration at $L = 24$, namely $((6, 6, 6, 6), \text{sh})$. In the framework of the present paper, this is the element

$$\pi_1 = (6, 6, 6, 6) \in \Pi(24), \quad H_1 = \text{sh}, \tag{44}$$

with $(\pi_1, H_1) \in \mathcal{C}(24)$.

Proposition 11.1. Paper 1's configuration (π_1, H_1) lies strictly in the interior of $\mathcal{C}(24)$, and strictly in the capability region $\mathcal{C}_{\text{cap}}(24)$.

Proof. $(\pi_1, H_1) \neq \hat{0}$ (since $|\pi_1| = 4 \neq 1$) and $(\pi_1, H_1) \neq \hat{1}$ (since $\min_k d_k = 6 > 1$, hence $\pi_1 \neq (1, \ldots, 1)$). Furthermore, $|\pi_1| = 4 \geq 2$ and $\min_k d_k = 6 \geq d_{\min}$ for any reasonable threshold ($d_{\min} = 2$ suffices per Paper 1's empirical results), so $(\pi_1, H_1) \in \mathcal{C}_{\text{cap}}(24)$. □

11.2 Paper 1 Findings as Properties of (π_1, H_1)

Paper 1 established the following properties of its configuration:

(F1) Bounded gap vs BP. At 1.3B parameters, BPB of (π_1, H_1) was 6.52% above BP.
(F2) WAND acceleration. $1.82\times$ wall-clock speedup with matched top-1 accuracy.
(F3) Prefix pruning. Stage 1 exit at 25% compute with 87.5% factual accuracy retention.
(F4) Parallel composition. Logit gain of $+2.4$ under log-space aggregation.
(F5) Dual-head SFT preservation. Bit-identical base retention ($\Delta\text{logit} = 0$).
(F6) Multi-specialist composition. Post-hoc specialist assembly without base retraining.
(F7) Apple Silicon verification. Speedups of $2.9\times$ and $2.7\times$ on MLX for Stage 1 and prefix pruning respectively.

In the present framework, these are properties of the specific configuration (π_1, H_1). They are not claims about BP or about fully local learning; they are claims about a particular point in $\mathcal{C}(24)$. The framework embeds these findings rather than supersedes them.

11.3 What the Framework Adds

The framework clarifies *why* the configuration (π_1, H_1) was a reasonable choice. It lies interior to $\mathcal{C}_{\text{cap}}(L)$, balancing:

- Substantial Stage 0 depth ($d_0 = 6 = L/4$) for representational work;
- Moderate stage count ($K = 4$) for capability properties;
- Shared head ($H = \text{sh}$) for parameter economy.

The framework also indicates *other* configurations that could have been chosen, each with a different property profile. A systematic comparison across the capability region is the empirical program sketched in §10.6.

Beyond the original $\mathcal{C}(24)$, the architecture-expanded space $\mathcal{C}^+(24)$ (§10.4) indicates that the structural features of Paper 1 that most contributed to its capability matrix — the dual-head SFT preservation, the parallel composition gain — are not specific to the chain, shared-head, full-CE point. They extend to stage forests (§10.5.2) and residual Log-OP objectives (§10.5.4), which are architecturally distinct and may weaken the $\sim 6.5\%$ BPB gap while preserving (F2)–(F7). This is the program's most actionable prediction for future iterations of Paper 1-style deployments.

11.4 No Invalidation

Remark 11.2. The findings of Paper 1 are not invalidated by the present work. They remain correct empirical claims about (π_1, H_1) and, in particular, about the 6.52% BPB gap at 1.3B scale. The present framework merely places these findings in a larger structural context.

---

12. Limitations of the Present Work

The present paper is substantially limited along the following axes.

(L1) Empirical scope. The empirical content is confined to Lemma 8.1, a one-dimensional slice measured by a single metric at a single scale with single-seed runs. Multi-seed replication, multi-metric measurement, and multi-scale validation are all required.

(L2) Structural scope. The framework $\mathcal{C}(L)$ is presented with its lattice structure and endpoint theorems. The categorical reformulation is only sketched (Appendix B). A full categorical treatment — including a formal *training functor* as a pseudofunctor $\mathcal{C}(L) \to \mathbf{TrainingOps}$, and natural transformations between it and external theories — is not developed.

(L3) Property formalization. The properties of §7 are defined informally. A rigorous formalization — e.g., via information-theoretic quantities for BP-descending properties, and categorical / operadic quantities for local-ascending properties — would strengthen the framework.

(L4) Connection to prior local-learning proposals. Prior local-learning proposals (target propagation, synthetic gradients, Forward-Forward, greedy layerwise) have not been placed systematically within $\mathcal{C}(L)$. Each likely corresponds to a specific configuration or to a modification of the framework (e.g., alternate loss structures beyond stop-gradient-separated cross-entropy).

(L5) Optimization-theoretic questions. The training operator $T(\pi, H)$ induces a gradient flow on $\Theta(\pi, H)$; the structure of this flow — its fixed points, its convergence properties, its stability — is not characterized. The relaxed operator over $\overline{\mathcal{C}}(L)$ (§9) is similarly uncharacterized.

(L6) Biological correspondence. The claim that local-ascending properties increase biological plausibility is asserted rather than proven. A rigorous biological correspondence would require a specific biological model (e.g., cortical columns, predictive coding) and a formal embedding.

(L7) Scaling. All empirical claims are at or below 362M parameters and $L = 12$. Behavior at the scales of Paper 1 (1.3B, $L = 24$) and beyond is extrapolated.

(L8) Configuration selection theory. The framework provides the space $\mathcal{C}(L)$ (and its extensions $\overline{\mathcal{C}}(L)$, $\mathcal{C}^+(L)$) but does not provide a guaranteed-optimal algorithm for selecting configurations for a given deployment. The multi-objective utility J of §10.1 is a framework, not a closed-form recipe.

(L9) Root bottleneck principle: informal proof status. Proposition 8.5 assumes a deterministic forward pass (no input noise, no stochastic layers). The extension to stochastic architectures (dropout, variational layers, MoE routing) is not formalized. Corollary 8.6 assumes the Markov chain $Y \to X \to r_\pi(x) \to h_{\ell_k(\pi)}$ is well-defined, which fails if forward randomness is not independent across stages.

(L10) Root bottleneck mitigations: untested at scale. §10.5's five mitigations (detached bus, forest, overlapping intervals, residual objective, multiplicative gate) are theoretically motivated but empirically untested within this paper. Experiments A–E of §10.7 propose their evaluation; none have been run.

(L11) Relaxed configuration space: hardening guarantees. §9's relaxation $\overline{\mathcal{C}}(L)$ makes λ learnable, but the hardening schedule (temperature annealing, §10.2.1) is a heuristic. Whether the hardened hard-vertex configuration is the one minimizing J — rather than the nearest hard vertex to the soft optimum — is not established.

(L12) Architecture-expanded space: no algebraic structure yet. $\mathcal{C}^+(L)$ is introduced combinatorially as (G, S, H, O, R) quintuples, without a lattice or categorical structure analogous to $\mathcal{C}(L)$'s refinement lattice. A proper algebraic treatment of $\mathcal{C}^+(L)$ is future work.

(L13) Bandwidth lens: scope and transferability. The answer bandwidth interpretation of §8.9 rests on Jeong (2026c), whose direct experiments are on CIFAR-10 (VGG CNN) and WikiText-2 (GPT-2); its conclusions are demonstrated at ≤ 124M parameters. Transfer to Paper 1's 1.3B Transformer regime is inferred, not measured. Experiment E of §10.7 is the minimal test; its result may sharpen or partially retract Proposition 8.16 in the Transformer regime (cf. Caveat 10.20 on residual-stream bandwidth compensation).

(L14) Bayesian evidence lens: scope. §7.6's reinterpretation of stage outputs as calibrated evidence estimators relies on Jeong (2026a, 2026b) for the calibration and Log-OP composition arguments. These works establish the framework in their own domains (information retrieval scoring; two-layer analytic derivation) but are not directly scaled to the 1.3B stage-partitioned regime. In particular, whether stage-local posteriors at Paper 1's scale are empirically well-calibrated — the premise of C_{cal} in Definition 10.1 — is an open measurement, not an established fact.

---

13. Conclusion

We have introduced the stage-partitioned learning family $\mathcal{C}(L)$, a combinatorial and algebraic configuration space that unifies backpropagation and fully local per-layer learning as the two extremal points of a bounded lattice. The intermediate region is populated by genuinely novel configurations that inherit, in systematically varying proportions, properties from both endpoints: representational coherence and end-to-end optimality from backpropagation; memory locality, modularity, biological plausibility, and the capability matrix of Paper 1 from local learning.

The framework's structural claims are sharpened by three subsequent results. First, the Stage 0 depth monotonicity observed empirically at $L = 12$ (§§8.1–8.6) is not a parochial finding but the empirical signature of a **root bottleneck principle** (§8.7): in a chain-partitioned model without independent input access to later stages, the first-stage representation is an information bottleneck for all later stages. The same monotonicity admits a sharper, capacity-theoretic reading as a **root bandwidth bottleneck** (§8.9), in which bounded boundary operators without multiplicative bypass additionally compress effective rank and thereby tighten the ceiling. Second, these principles imply the **BPB collapse proposition** (§8.8): BPB is a BP-descending projection of $\mathcal{C}(L)$; BPB-only optimization converges to the BP endpoint. Third, the correct formulation of dynamic configuration selection is therefore a **multi-objective Pareto problem** (§10.1) over BPB, locality/modularity, capability-matrix utility, bandwidth, and calibration — not a BPB minimization.

Two lenses from prior work anchor the framework to a larger theoretical setting. The **Bayesian evidence lens** (§7.6), drawing on Jeong (2026a, 2026b), reinterprets stage outputs as calibrated evidence estimators and stage aggregation as Logarithmic Opinion Pooling — making Paper 1's WAND, prefix pruning, and parallel-composition findings instances of a normative Bayesian combination rule rather than heuristic ensembles. The **answer bandwidth lens** (§8.9), drawing on Jeong (2026c), supplies the capacity-theoretic complement: a boundary operator that is also a bounded map destroys representational rank, independently of gradient flow. The two lenses are orthogonal — calibration and bandwidth are distinct properties — and both must be tracked at every boundary.

To make these statements quantitative we relax $\mathcal{C}(L)$ to a continuous space $\overline{\mathcal{C}}(L) = [0, 1]^{L-1} \times \mathbf{H}$ (§9) of soft detachment coefficients, and extend to $\mathcal{C}^+(L) = (G, S, H, O, R)$ (§10.4) allowing non-chain stage graphs, independent input sources, alternative objective families, and alternative boundary relay structures. Five structural mitigations of the root bottleneck are identified (§10.5): detached input buses, stage forests, overlapping interval experts, residual Log-OP objectives, and bandwidth-preserving multiplicative boundary gates. The residual Log-OP objective (Experiment A) and the bandwidth diagnostic (Experiment E1) are the two highest-information experiments at lowest implementation cost (§10.7).

The principal statements of the paper can be summarized as:

> BPB is not a neutral metric over $\mathcal{C}(L)$. It is a BP-descending projection.

> Stage 0 depth dominance is the empirical signature of a root bottleneck: in a chain-partitioned model, all later stages are functions of the first-stage representation, and therefore cannot recover target-relevant information discarded by the root. When the boundary operator is bounded without multiplicative bypass, this bottleneck is compounded by a bandwidth bottleneck: effective rank is compressed, tightening the ceiling further.
>
> Dynamic boundary selection must be formulated as Pareto selection, not BPB minimization. BPB minimization alone collapses the configuration toward the BP endpoint.
>
> To obtain interior configurations competitive on BPB without surrendering capability-matrix properties, one must expand the architecture beyond chain partitions with identity relays: detached input buses, parallel stage forests, overlapping interval experts, residual Log-OP objectives, and bandwidth-preserving multiplicative boundary gates are the natural extensions.
>
> *Detachment is a gradient operation, not an information-deletion operation.*

These statements admit a single integrating proposition that unifies the present framework with the three prior works of the author:

> Stage-partitioned learning is not merely a partition of gradient flow. It is a partition of Bayesian evidence construction. For the partition to remain useful, each boundary must preserve enough answer bandwidth for downstream stages to contribute independent evidence. When the first detached representation is bandwidth-insufficient, BPB pulls the configuration toward deeper Stage 0 and ultimately toward BP.

The standard dichotomy of *BP versus local learning* is dissolved; it becomes a *choice among points* in a unified space, structured simultaneously by gradient topology (the $\mathcal{C}(L)$ lattice), by Bayesian evidence composition (the Log-OP aggregation rule), and by answer bandwidth (the relay structure R). Which point is selected depends on which properties one values. No single endpoint is universally optimal; nor is any single interior point. The optimal configuration is deployment-specific, and the framework — together with its relaxation $\overline{\mathcal{C}}(L)$ and its expansion $\mathcal{C}^{+}(L)$ — provides the language for that optimization.

The prior work of Jeong (2026d) corresponds to one specific, and empirically reasonable, point in this space. The present framework clarifies what that correspondence is, what its structural preconditions are, and what else is possible.

References

Belilovsky, E., Eickenberg, M., & Oyallon, E. (2019). Greedy layerwise learning can scale to ImageNet. *Proceedings of ICML*.

Broder, A. Z., Carmel, D., Herscovici, M., Soffer, A., & Zien, J. (2003). Efficient query evaluation using a two-level retrieval process. *Proceedings of CIKM*.

Grossberg, S. (1987). Competitive learning: From interactive activation to adaptive resonance. *Cognitive Science*, 11(1), 23–63.

Hinton, G. E. (2002). Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8), 1771–1800.

Hinton, G. E. (2022). The Forward-Forward algorithm: Some preliminary investigations. *arXiv:2212.13345*.

Jaderberg, M., Czarnecki, W. M., Osindero, S., Vinyals, O., Graves, A., Silver, D., & Kavukcuoglu, K. (2017). Decoupled neural interfaces using synthetic gradients. *Proceedings of ICML*.

Jeong, J. (2026a). Bayesian BM25: A Probabilistic Framework for Hybrid Text and Vector Search. *Zenodo*. https://doi.org/10.5281/zenodo.18414940

Jeong, J. (2026b). From Bayesian Inference to Neural Computation: The Analytical Emergence of Neural Network Structure from Probabilistic Relevance Estimation. *Zenodo*. https://doi.org/10.5281/zenodo.18512411

Jeong, J. (2026c). Answer Bandwidth: Why Sigmoid Fails in Hidden Layers. *Zenodo*. https://doi.org/10.5281/zenodo.19254501

Jeong, J. (2026d). Product of Experts as Scalable Local Learning: Modular Construction at 1.3B Parameters (v5). *Zenodo*. https://doi.org/10.5281/zenodo.19547653

Lee, D.-H., Zhang, S., Fischer, A., & Bengio, Y. (2015). Difference target propagation. *Proceedings of ECML-PKDD*.

Lillicrap, T. P., Cownden, D., Tweed, D. B., & Akerman, C. J. (2016). Random synaptic feedback weights support error backpropagation for deep learning. *Nature Communications*, 7, 13276.

Löwe, S., O'Connor, P., & Veeling, B. S. (2019). Putting an end to end-to-end: Gradient-isolated learning of representations. *Proceedings of NeurIPS*.

Ma, W.-D., Lewis, J. P., & Kleijn, W. B. (2020). The HSIC bottleneck: Deep learning without backpropagation. *Proceedings of AAAI*.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323(6088), 533–536.

Whittington, J. C. R., & Bogacz, R. (2019). Theories of error back-propagation in the brain. *Trends in Cognitive Sciences*, 23(3), 235–250.

Appendix A. Proofs

A.1 Proof of Proposition 5.2

Statement. Under $(\pi, H) = ((L), \mathrm{ps})$, the training operator is equivalent to BP with overparameterized head $W' = W_{\mathrm{hd}} + W_{\mathrm{hd},0}$.

Proof. At $K = 1$, there is a single stage with logit $z_0 = (W_{\mathrm{hd}} + W_{\mathrm{hd},0}) \cdot h_L = W' \cdot h_L$. The modified loss is $\tilde{\mathcal{L}}^{(L),\mathrm{ps}} = \mathrm{CE}(W' h_L, y)$, which has no sg operations (single stage implies no internal boundaries).

For the hidden-parameter gradients: the forward computational graph is $(\theta^{\mathrm{hid}}) \to h_L \to W' \to \mathrm{CE}$. Reverse-mode differentiation yields

$$\frac{\partial \tilde{\mathcal{L}}^{(L),\mathrm{ps}}}{\partial \theta_i} = \frac{\partial \mathrm{CE}(W' h_L, y)}{\partial h_L} \cdot \frac{\partial h_L}{\partial \theta_i} \tag{45}$$

for any hidden parameter θ_i. Under standard BP with a single head W', the right-hand side is $\partial \mathcal{L}^{\mathrm{BP}} / \partial \theta_i$, so the two gradients are equal.

For the head gradients: $W' = W_{\mathrm{hd}} + W_{\mathrm{hd},0}$ depends linearly on each of W_{hd} and $W_{\mathrm{hd},0}$, hence

$$\frac{\partial W'}{\partial W_{\mathrm{hd}}} = \frac{\partial W'}{\partial W_{\mathrm{hd},0}} = I, \tag{46}$$

and by the chain rule

$$\frac{\partial \tilde{\mathcal{L}}^{(L),\mathrm{ps}}}{\partial W_{\mathrm{hd}}} = \frac{\partial \tilde{\mathcal{L}}^{(L),\mathrm{ps}}}{\partial W_{\mathrm{hd},0}} = \frac{\partial \mathrm{CE}(W' h_L, y)}{\partial W'}. \tag{47}$$

Thus the gradients on the two head matrices are equal. Under gradient descent with equal learning rates, the two matrices evolve in the same direction. The sum W' evolves consistently with standard BP on a single head. □

A.2 Proof of Theorem 4.9

Statement. Under refinement $(\pi, H) \sqsubseteq (\pi', H')$, (i) forward values of $\mathcal{L}^{\pi,H}$ are recovered as a partial sum of those of $\mathcal{L}^{\pi',H'}$; (ii) gradient structures differ by additional stop-gradient operations in the finer configuration.

Proof sketch.

(i) Let $\pi' = (d'_0, \dots, d'_{K'-1})$ refine $\pi = (d_0, \dots, d_{K-1})$, so the boundary set of π is a subset of that of π'. There is a well-defined map $\phi : \{0, \dots, K'-1\} \to \{0, \dots, K-1\}$ sending a stage of π' to the containing stage of π. The forward value of $\mathcal{L}^{\pi,H}$ involves only the termini of π, which are a subset of the termini of π'. Hence the forward logits at stages of π are computed from the same hidden states as those of π', up to head-structure parameterization. The forward sum over stages of π therefore equals a partial sum over stages of π'.

(ii) The modified loss $\tilde{\mathcal{L}}^{\pi',H'}$ contains one sg operator per internal boundary of π'. The modified loss $\tilde{\mathcal{L}}^{\pi,H}$ contains one sg operator per internal boundary of π. Since the boundaries of π are a *subset* of those of π', the finer configuration $\tilde{\mathcal{L}}^{\pi',H'}$ has *additional* sg operators that $\tilde{\mathcal{L}}^{\pi,H}$ does not have. These additional sg operators zero additional gradient contributions in the finer configuration. Hence the gradient structures differ. □

A.3 Proof of Proposition 4.7

Statement. Under (π, H), the gradient on stage k's hidden parameters depends only on the k-th loss summand.

Proof. Consider a loss summand with index j. Its forward-pass computation involves the hidden states h_i for $\ell_{j-1}(\pi) < i \le \ell_j(\pi)$. For $j < k$: these hidden states are all computed before stage k begins, so θ_k^{hid} does not participate in the j-th forward pass; hence $\partial \tilde{\mathcal{L}}_j / \partial \theta_k^{\mathrm{hid}} = 0$. For $j > k$: the forward pass through stages $k+1, \dots, j$ begins from $\mathrm{sg}(h_{\ell_k(\pi)})$, so θ_k^{hid} contributes to $h_{\ell_k(\pi)}$ but that contribution is zeroed by sg in the backward pass. Hence $\partial \tilde{\mathcal{L}}_j / \partial \theta_k^{\mathrm{hid}} = 0$ for $j > k$ as well. Only $j = k$ contributes, establishing the

claim. □

Appendix B. Categorical Reformulation

A more abstract formulation of $\mathcal{C}(L)$ places it within a category-theoretic framework.

B.1 The Category of Configurations

Define a category $\mathbf{C}(L)$ whose objects are configurations $(\pi, H) \in \mathcal{C}(L)$ and whose morphisms are refinements: there is a unique morphism $(\pi, H) \to (\pi', H')$ iff $(\pi, H) \sqsubseteq (\pi', H')$. This presents $\mathcal{C}(L)$ as a *thin category* whose underlying poset is the configuration lattice.

B.2 The Parameter-Space Fibration

Define a functor $\Theta : \mathbf{C}(L)^{\mathrm{op}} \to \mathbf{Set}$ sending a configuration to its parameter space and a refinement morphism $(\pi, H) \to (\pi', H')$ to the inclusion $\Theta(\pi, H) \hookrightarrow \Theta(\pi', H')$. This is contravariant because a finer configuration has a larger parameter space.

B.3 The Training-Operator Family

The training operators $\{T(\pi, H)\}_{(\pi,H)\in\mathcal{C}(L)}$ form a *family of sections* of the parameter-space fibration: each $T(\pi, H)$ is a gradient-descent map on $\Theta(\pi, H)$. They are *not* natural in the categorical sense: a refinement morphism does not induce a compatible morphism between training operators, because of the additional sg operators (Theorem 4.9(ii)).

A formal treatment would construct a *weighted category* or *operadic structure* in which the training operators are morphisms, and would identify the "discontinuity" at stage boundaries as an additional piece of structure beyond the lattice.

B.4 Open Categorical Questions

- Does $\mathcal{C}(L)$ admit a natural ∞-categorical enhancement in which the training dynamics live at the $(n+1)$-level?
- Is there a universal property of BP as the minimum element (e.g., as a limit of some functor)?
- Do the endpoint identifications of §§5, 6 arise from a natural transformation between T and external theories of BP and Forward-Forward?

These questions are deferred.