TRANSMITTER: SIGNAL-TRANSDUCTION-INSPIRED REUSABLE RESPONSE PATHWAYS THROUGH DECOUPLED DISCRETE STATE SELECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Biological signaling provides a role-level analogy for separating response selection from response expression. Inspired by this principle, we introduce *Transmitter*, a vision backbone whose token-mixing operator makes three roles explicit: query fragments select a route in a learned receptor book, the route retrieves a paired vector from a signal book, and an input-dependent gain modulates the retrieved response. The assembled response is added to each head input, exposing route and response traces for fixed-checkpoint analysis. During training, the model learns both which response to select and how to express it for classification. The classification loss trains the selected response path, while an auxiliary loss learns the query-to-receptor matching used for discrete selection. At one frozen intermediate checkpoint, replacing the release gain of all eight blocks by its within-image spatial mean lowers ImageNet-1K top-1 accuracy by 21.58 percentage points (pp), while moving whole per-token gain vectors to other tokens lowers it by 16.18 pp; forcing the original routes leaves drops of 20.46 and 15.60 pp, so gain placement matters even without route reselection. In exploratory COCO traces, the same-minus-different-class code-affinity gap is 0.0078 larger in the last two blocks than in the first two. These results position route-explicit token mixing as an inspectable decomposition of visual computation.

1 INTRODUCTION

Modern vision backbones learn continuous representations, while self-attention computes query-conditioned weights over transformed token values (Vaswani et al., 2017). Its parameterization couples addressing, response synthesis for every context token, and output mixing. Pairwise interaction and redundant token computation drive the cost of vision transformers, motivating token pruning and sparse attention (Pan et al., 2021; Rao et al., 2021; Zhu et al., 2023). These methods reduce interactions but leave a basic question open: must response identity be regenerated from every token? We ask whether token mixing can retain query-dependent selection while storing response identity in a learned, reusable substrate.

Multi-query and grouped-query attention share key/value heads across query groups, reducing KV-cache traffic and projection parameters (Shazeer, 2019; Ainslie et al., 2023). Persistent-memory attention and discrete key-value bottlenecks likewise separate learned response parameters from token states (Sukhbaatar et al., 2019; Träuble et al., 2023). These results motivate an asymmetric factorization: the query remains input-dependent while response identity is carried by a learned substrate.

Transduction tests this factorization in a vision backbone. After head grouping, an input-dependent query fragment selects a receptor-book row by nearest-row assignment and retrieves its paired signal-book vector. Writing f_Q for the query transformation, the operation is

$$q_i = f_Q(x_i), \qquad k_i^\star = \arg\min_k \|q_i - m_k\|_2^2, \qquad r_i = g_i v_{k_i^\star}. \tag{1}$$

The assignment forms vector-quantized Voronoi cells in the local query space under the auxiliary objective (van den Oord et al., 2017). A finite response substrate indexed by K creates a conditional compactness hypothesis: the complete operator is smaller for appropriate codebook, query, signal, and strength dimensions.

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

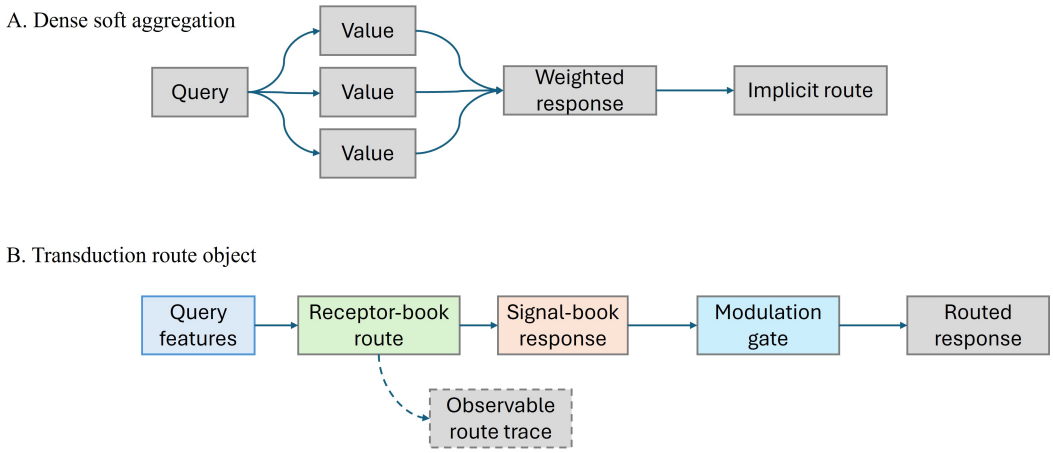


Figure 1: **Dense aggregation versus explicit route-object computation.** (A) Standard attention forms a soft distribution over value vectors and collapses it into a weighted response, leaving no separate learned route object for later reuse. (B) Transduction separates query-conditioned receptor-book route selection, signal-book response retrieval, and input-dependent modulation. The exposed route and gain traces reveal gain-sensitive predictions, stronger within-token pattern effects in matched-size comparisons, and class-related code affinity across depth.

A biological analogy suggests why these roles may be separated. At a synapse, a presynaptic terminal releases neurotransmitters; ligand–receptor recognition, functionally analogous to a docking event, selects a downstream signaling pathway, while intracellular mechanisms regulate response amplitude (Abbott & Regehr, 2004; Tsodyks & Markram, 1997; Sheng & Kim, 2002). Transduction uses the same role separation: an input-dependent query selects a receptor/response entry, and a separate gain controls how strongly its signal is expressed.

Transduction exposes route and gain traces during forward inference, enabling fixed-checkpoint interventions and route diagnostics. Unlike discrete key-value bottlenecks on a frozen encoder, it places address–response separation inside trainable token-mixing layers and adds the modulated response to the token stream (Träuble et al., 2023); the biological boundary is given in Appendix B.2.

We evaluate Transmitter at three levels. ImageNet-scale registry experiments test predictive utility; controlled ablations compare soft response mixing, direct key-value lookup, and hard discrete-code emission. Gain interventions demonstrate strong prediction sensitivity to response modulation, and matched-size decomposition shows greater sensitivity to within-token gain patterns than token-level changes, while code-affinity and exploratory gain-profile diagnostics track class-related structure across depth. The analyses appear in Section 4.4 and Appendices E.1 and E.3.

2 Related Work

Attention and shared key-value state. Self-attention computes query-conditioned weights over value vectors and is a central operator in sequence and vision models (Vaswani et al., 2017; Dosovitskiy et al., 2021). Modern Hopfield analyses connect this operation to associative retrieval (Ramsauer et al., 2021), while multi-query and grouped-query attention share key-value state across heads or groups (Shazeer, 2019; Ainslie et al., 2023). Attention can also incorporate learned, input-independent key-value memory (Sukhbaatar et al., 2019). These methods return weighted responses. Transduction instead hard-selects a receptor-book entry, retrieves its paired signal-book response, and exposes the selected route and modulation strength as separate logged objects.

Discrete latents and addressable memory. Vector-quantized models show that continuous inputs can be committed to learned discrete codes, and the selected code is itself the emitted representation (van den Oord et al., 2017). Key-value memory networks, product-key memory, and large-scale memory layers separate learned addresses from retrieved content (Miller et al., 2016; Lample et al., 2019; Berges et al., 2025). Closest to our hard selection rule, the discrete key-value bottleneck initializes keys from pre-trained encoder features with exponential moving averages, freezes them for class-incremental learning, and passes the paired retrieved values to a downstream decoder (Träu-

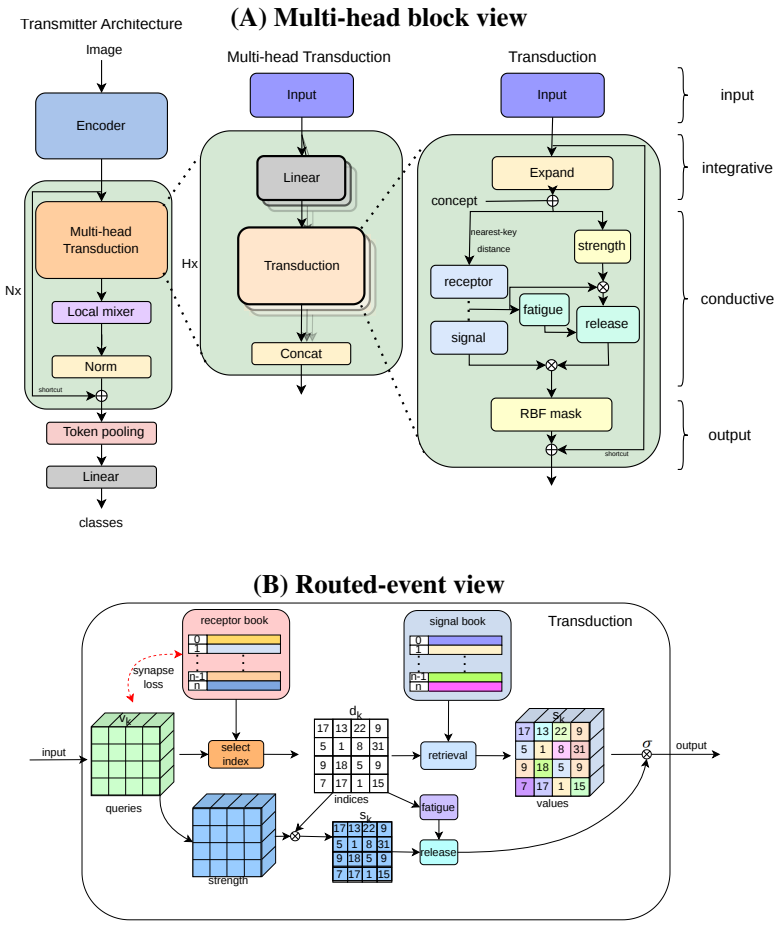


Figure 2: **Transduction operator overview.** (A) Inside a Transmitter block, multiple heads perform hard nearest-row assignment in the receptor book (the matching table) and produce routed responses in parallel. (B) At the routed-event level, query fragments select route codes, retrieve paired signal-book response vectors, and express them through input- and usage-dependent modulation before contributing to the output. The selected route codes are recorded as forward-pass traces for later analysis.

ble et al., 2023). Transduction instead shapes its receptor book with auxiliary objectives inside a vision token-mixing layer and modulates the retrieved signal-book response before residual addition. Transduction differs in its address update, response-expression path, and role within the backbone.

Routing. Routing is input-dependent selection of reusable computation or parameters. Conditional computation studied sparse stochastic gates (Bengio et al., 2013); mixture-of-experts scales routing to expert branches (Shazeer et al., 2017; Fedus et al., 2022; Riquelme et al., 2021). In these MoE layers, learned gates route tokens to expert functions and weight their computed outputs. Transduction instead hard-selects a receptor-book entry, retrieves its paired learned signal response, and modulates that response inside each head, exposing the selected route and gain.

3 METHOD

Transduction separates query-based selection from response retrieval and modulation. An input-dependent query selects an entry in a learned receptor book, retrieves the paired signal-book vector, and modulates its contribution through an expression gain. The Transmitter backbone incorporates this operator into a multi-head vision architecture built from a convolutional stem and repeated Transmitter blocks. Its receptor book, signal book, and expression-gain terms make the three computational roles explicit. Figure 2 shows the two scales used below: block-level multi-head routing and event-level route selection, response retrieval, and modulation.

Notation. The main text uses two notation levels. At the block level, $X_b^{(\ell)} \in \mathbb{R}^{C\times L}$ is the token state for image b at block ℓ, with C channels and L spatial tokens. The network has N blocks, each with H heads. Inside block ℓ and head h, the feature space is organized into S parallel response factors and J channel chunks. Let $p = (\ell, h, s, j)$ denote a fixed factor–chunk pathway, and let $t \in \{1, \dots, L\}$ index a spatial token. For a fixed image b, a routed event $i = (p, t)$ denotes one local selection decision. The receptor book $M = \{m_k\}_{k=1}^K \subset \mathbb{R}^{d_q}$ is learned at the block/head module level, from which each event independently selects a route id $k_i^\star \in \{1, \dots, K\}$. Signal-book vectors $v_k \in \mathbb{R}^{d_v}$, retrieved responses $\bar{v}_i, r_i \in \mathbb{R}^{d_v}$, and scalar gains g_i are parameterized per pathway p. Thus the full response-path identity is $(p, k_i^\star)$, while a receptor id alone is not a globally shared route label. The notation suppresses p and t inside event-level equations; forward traces retain $b, k_i^\star, g_i$, with p and t available in the extended logs.

Hybrid tokenization and block structure. Let I_b be the b-th image. A convolutional stem E_ϕ first maps the image to a feature grid with height H_s and width W_s. The grid is then flattened over spatial positions into $L = H_s W_s$ tokens and augmented with a learned spatial embedding:

$$\begin{aligned} F_b &= E_\phi(I_b) \in \mathbb{R}^{C\times H_s\times W_s}, \\ X_b^{(0)} &= \mathrm{flat}_{hw}(F_b) + P \in \mathbb{R}^{C\times L}. \end{aligned} \tag{2}$$

Here E_ϕ is the convolutional stem, flat_{hw} reshapes the $H_s \times W_s$ grid into L spatial tokens, and $P \in \mathbb{R}^{C\times L}$ is the learned spatial embedding. The model contains N Transmitter blocks. Each block applies head splitting, per-head Transduction, head concatenation, and a local convolutional mixer; no projection is applied before the heads are split. Since those steps are implemented directly in the released code, the main text writes the block as a compact residual update; these surrounding operations do not introduce additional route identities. Let B_ℓ denote the full routed block body shown in Figure 2A, and let split_h select the h-th head slice:

$$B_\ell(X) = \mathcal{M}_\ell\left(\mathrm{Concat}_{h=1}^H \mathcal{T}_{\ell,h}\left(\mathrm{split}_h(X)\right)\right), \tag{3}$$

Here $W_\ell X + \beta_\ell$ has shape $C \times L$, with the bias broadcast over spatial tokens; the head slices partition the C channels, and $\mathcal{M}_\ell$ returns a $C \times L$ representation. Thus the residual addition below is dimensionally well-typed. The block output is

$$X_b^{(\ell+1)} = \mathrm{LayerNorm}\Big(X_b^{(\ell)} + \mathrm{StochasticDepth}_{p_\ell}(B_\ell(X_b^{(\ell)}))\Big). \tag{4}$$

Here p_ℓ is the stochastic-depth probability for block ℓ; its schedule is specified with the training recipe in Appendix D.1.

Hard route selection and receptor-book training. Figure 2B zooms in on one routed event. Each route entry k has two learned rows: a receptor-book row $m_k \in \mathbb{R}^{d_q}$ and a signal-book row $v_k \in \mathbb{R}^{d_v}$. The query fragment q_i is taken directly from the grouped head features, without a learned query projection or query normalization. Route selection is nearest-row lookup in the receptor book, with ties resolved to the lowest index:

$$k_i^\star = \arg\min_{k\in\{1,\dots,K\}} \|q_i - m_k\|_2^2, \qquad \bar{v}_i = v_{k_i^\star}. \tag{5}$$

The selected route id is logged as $k_i^\star$. The emitted response is not the receptor-book row itself; it is the paired signal-book row $\bar{v}_i$, later scaled by an input-dependent gain. This is the main distinction from standard single-codebook vector quantization (van den Oord et al., 2017), where the selected code is also the emitted representation.

The hard index is not relaxed into a soft routing distribution. During training, the receptor book and query geometry are shaped by a VQ-style stop-gradient auxiliary objective:

$$\mathcal{L}_{\mathrm{route},\ell,h} = \lambda_{\mathrm{commit}}\mathcal{L}_{\mathrm{VQ},\ell,h} + \mathcal{L}_{\mathrm{align},\ell,h}. \tag{6}$$

$\lambda_{\mathrm{commit}} \geq 0$ weights the VQ-style term. The VQ term contains key-side and query-side squared-distance updates for the selected receptor-book row, with entropy-weighted usage terms to discourage early route collapse. The alignment term regularizes grouped query fragments with a centered cosine objective. These losses are used only during training; normal inference still uses the hard lookup in Eq. 5. Appendix C.2 gives the implementation-facing objective.

Fatigue-style response modulation. The signal-book response is not emitted with a fixed amplitude. Transduction uses a fatigue-style gate as a role-level abstraction of use-dependent modulation in synaptic transmission, where finite releasable vesicle pools can be depleted and replenished with activity (Rizzoli & Betz, 2005; Wang & Kaczmarek, 1998). In the implemented operator, repeated selection of the same route changes how strongly its signal-book response is expressed. For image b in the current block/head, let $\mathcal{I}_b$ be its routed events across the S response factors, J channel chunks, and L spatial tokens. The route-usage fraction is

$$\pi_{b,k} = \frac{1}{|\mathcal{I}_b|} \sum_{i \in \mathcal{I}_b} \mathbf{1}(k_i^\star = k). \tag{7}$$

where $\mathbf{1}(\cdot)$ is the indicator function. All route-indexed quantities in this paragraph are local to the current block/head. For one event i, the response amplitude is determined by three quantities: the learned scalar response-scale parameter $a_{k_i^\star}$ of the selected route, an event-level strength σ_i, and the current usage fraction $\pi_{b,k_i^\star}$. Let x_i be the event feature from the same routed event that produced q_i, and let $S_{\ell,h}^{(p)}$ denote the strength-projection slice for the event's pathway. The selected strength is

$$s_i = S_{\ell,h}^{(p)}(x_i) \in \mathbb{R}^K, \qquad \sigma_i = s_{i,k_i^\star}, \tag{8}$$

where s_i contains one predicted strength for each route; σ_i is clipped to $[-20, 20]$ before the gate. The scalar gain and modulated response are

$$g_i = \mathrm{softplus}(a_{k_i^\star}) \, \mathrm{sigmoid}(\sigma_i - \pi_{b,k_i^\star}), \qquad r_i = g_i \, \bar{v}_i. \tag{9}$$

The modulated response r_i is the event-level value passed to output assembly. Appendix C.1 gives implementation-facing pseudocode for this fatigue-style gate.

Output assembly. After modulation, the responses r_i of one token are reassembled over chunks into a response-factor array $R_t \in \mathbb{R}^{S \times d}$ for token t, where S is the number of response factors and d is the head width. A layer normalization is applied jointly over all $S \times d$ entries of each token, the factors are summed, and the result is added to the head input x_t:

$$z_t = \mathrm{LayerNorm}_{S \times d}(R_t), \qquad \mathcal{T}_{\ell,h}(x)_t = x_t + \sum_{s=1}^{S} z_{t,s}. \tag{10}$$

Because the normalization is taken within each token, a common rescaling of all gains of a token is largely removed, whereas the relative pattern of gains across the chunks and response factors of that token is retained. This reshape-and-combine step changes feature values but not route identities: the logged trace fields remain $k_i^\star$ and g_i.

Trace logging. During normal forward inference, forward hooks record each routed event, keyed by its event index i, as $\tau_i = (\ell, h, b, k_i^\star, g_i)$, where $k_i^\star$ is the selected route and g_i is its expression gain. Extended logs additionally store token, response-factor, and chunk coordinates for spatial or factorwise diagnostics.

Classifier. After the final block, per-channel softmax pooling over the L spatial tokens summarizes the features into $u_b \in \mathbb{R}^C$; equivalently, each $u_{b,c}$ is a softmax-weighted average of $X_{b,c,1:L}^{(N)}$ over spatial positions. These pooling weights are computed within each feature channel and are not class probabilities. A linear head then produces class logits $\hat{y}_b = W_{\mathrm{cls}} u_b + \beta_{\mathrm{cls}}$. For a minibatch with predictions $\hat{Y}$ and labels Y, training minimizes cross-entropy plus the route regularizers summed over all block/head pairs:

$$\mathcal{L}_{\mathrm{total}} = \mathcal{L}_{\mathrm{CE}}(\hat{Y}, Y) + \sum_{\ell=0}^{N-1} \sum_{h=1}^{H} \mathcal{L}_{\mathrm{route},\ell,h}. \tag{11}$$

4 Experiments

Transduction makes routing a first-class operation inside a vision backbone: queries select receptor-book routes, retrieve decoupled signal-book responses, and express those responses with input-dependent strength. We evaluate this design through the compute-matched ImageNet registry,

Table 1: **ImageNet-scale registry.** Clean top-1 accuracy, parameter count, and FLOPs for models trained from scratch under calibrated baseline configurations (↓: lower is better, ↑: higher is better). Transmitter entries are the physically-pruned `all_off`, H=4, 8-block, R=16 models. Top-1 is final-validation (epoch-1000) accuracy; baseline accuracies average two runs. FLOPs use the PyTorch `FlopCounterMode` counter. The dashed line separates the sub-10M parameter tier; bold marks the best value within each tier.

Model	Params (M) ↓	FLOPs (G) ↓	Top-1 (%) ↑
ViT-Tiny	**5.717**	2.150	53.161
Transmitter-512	5.808	**1.622**	61.154
ConvNeXtV2-Pico	9.066	2.734	**61.328**
Transmitter-768	**10.221**	**2.923**	63.094
ResNet-18	11.690	3.628	59.579
ViT-Small	22.051	8.482	57.477
ResNet-50	25.557	8.178	**63.319**
ConvNeXtV2-Tiny	28.635	8.911	63.104

ImageNet-100 operator ablations, and frozen epoch-610 analyses. The registry tests compactness at ImageNet scale, and the ablations compare the route-to-response path with simpler routing and code-emission alternatives. Gain interventions and matched-size decomposition test prediction sensitivity to gain placement and within-token allocation; COCO code-affinity, random-initialization controls, and gain-profile diagnostics examine structure across depth. CIFAR results provide smaller-scale context in the appendix.

4.1 Protocol and Metrics

The main registry uses from-scratch training and a shared evaluation recipe. Non-Transmitter entries are instantiated from standard `timm` architecture definitions (Wightman, 2019); all entries use the ImageNet-1K evaluation pipeline, while training recipes differ across model families. The registry reports clean ImageNet-1K top-1 accuracy, parameter count, and FLOPs. The separately recorded robustness evidence includes ImageNet-V2 (Recht et al., 2019), ImageNet-A under the 1000-class protocol (Hendrycks et al., 2021), and ImageNet-C mean corruption error (mCE). Following the ImageNet-C convention (Hendrycks & Dietterich, 2019), mCE is normalized by the Hendrycks-Dietterich AlexNet top-1 corruption-error reference over 15 corruptions and five severities. Detailed robustness evaluations on ImageNet-V2, ImageNet-A, and ImageNet-C are reported in Appendix D.2. Appendices D and F provide protocol cards and implementation notes.

The main registry uses the full ImageNet-1K validation split and the standard ImageNet-V2, ImageNet-A, and ImageNet-C evaluation sets. Clean accuracy, corruption results, parameter count, and FLOPs calibrate ImageNet-scale behavior and compactness under the recorded recipes. Operator ablations test which parts of Transduction matter under the ImageNet-100 recipe. The fixed-checkpoint analyses then intervene on the response gain of a frozen checkpoint and measure how the gain and the selected codes are organized across depth.

4.2 ImageNet Registry and Reduced-Scale Diagnostics

Table 1 reports clean accuracy, parameter count, and nominal FLOPs (measured via PyTorch `FlopCounterMode`) to evaluate whether the route-explicit design remains compact at ImageNet scale. Transmitter-512 has 5.808M parameters and 1.622G nominal FLOPs; its final-validation top-1 is 61.154%, compared with 53.161% for ViT-Tiny and 61.328% for ConvNeXtV2-Pico. Transmitter-768 has 10.221M parameters and 2.923G nominal FLOPs; its final-validation top-1 is 63.094%, compared with 63.319% for ResNet-50 and 63.104% for ConvNeXtV2-Tiny. These measurements evaluate parameter and FLOP efficiency at ImageNet scale under a compute-matched from-scratch training protocol, establishing the predictive feasibility of explicit routing in a standard vision backbone.

Appendix D.4 provides CIFAR-10/100 results; Table 1 is the ImageNet-scale registry used for the main comparison. Appendix D.5 reports a paired CIFAR training-log diagnostic: validation accuracy

Table 2: **ImageNet-100 routing-operator comparison.** The surrounding encoder, data split, resolution, optimizer, and training recipe are fixed; only the routing/response operator family is changed. Entries are final-validation (epoch-1000) top-1 accuracy averaged over three seeds.

Operator variant	Top-1 (%) ↑
Full Transduction routing	70.773
Soft key-value retrieval	69.507
Strict vector quantization	69.533
Hard key-value lookup	67.187

minus recorded training accuracy, where recorded training accuracy denotes the value logged during the training loop rather than a separate eval-mode pass over the training set. Transmitter-256 runs retain positive recorded gaps for longer intervals than the matched baselines in those logs; the small-model curves and paired counts are reported in Figure 4 and Table 8. This behavior reflects optimization under stochastic regularization; data augmentation, stochastic depth, and evaluation modes affect the recorded accuracies.

4.3 Operator Ablations

The controlled operator ablation uses ImageNet-100 as a reduced but nontrivial setting for isolating the route-to-response operation. The data split, surrounding encoder, resolution, optimizer, and training recipe are fixed, while only the route-to-response operation varies.

Full Transduction decouples hard nearest-receptor route selection from value-side signal response retrieval, then expresses the retrieved response through usage-dependent gain and the factorwise RBF mask. Soft key-value retrieval replaces hard commitment with soft key weights and a weighted value mixture. Strict vector quantization keeps a discrete assignment but emits the selected code itself rather than a separately parameterized signal response. Hard key-value lookup keeps hard nearest-key selection but removes the decoupled signal-response expression path.

The gain-and-mask pair separates two response-expression roles. The sigmoid gate follows the vesicle-pool and release-probability intuition: the selected signal-book row has an available amplitude, and each routed event releases a bounded fraction of that amplitude (Rizzoli & Betz, 2005; Zucker & Regehr, 2002). The factorwise RBF mask follows the tuning-curve intuition: expression is strongest near an effective factor state and decays smoothly as the state moves away (Sériès et al., 2004). Together, the two terms separate how much response is released from whether the current factor state is receptive to that response.

Table 2 highlights the dominant separation. Full Transduction routing reaches 70.773%. Variants that simplify the emitted response object fall farther: strict vector quantization emits the selected code itself and reaches 69.533%, while hard key-value lookup keeps hard route selection but removes the response-expression path and reaches 67.187%.

The largest descriptive accuracy contrast is between Full Transduction routing and hard key-value lookup: Full Transduction routing reaches 70.773%, while hard key-value lookup reaches 67.187%, a 3.586 percentage-point gap between the full response-expression path and a variant that keeps route selection but removes the decoupled signal response and its input-dependent expression. The 3.586-point accuracy gap is a contrast between configurations that jointly change gain, mask, and response decoupling. Soft key-value retrieval reaches 69.507%, comparable to strict vector quantization. Hard receptor selection supplies the countable routes and gains examined in the fixed-checkpoint analyses below.

4.4 Fixed-Checkpoint Gain Interventions and Trace Diagnostics

After training, we ask how the frozen model uses the response gain g_i of Eq. 9. All analyses in this subsection use one frozen epoch-610 checkpoint from one warm-start training trajectory, with parameters, readouts, and thresholds held fixed during inference.

Table 3: **Full-magnitude response-gain interventions at a frozen checkpoint.** ImageNet-1K validation (50,000 images) top-1 accuracy of one frozen epoch-610 checkpoint when the release gain g_i of all eight blocks is replaced by its within-image spatial mean, or when whole per-token gain vectors are shuffled across tokens (five permutations). *Free routes*: downstream routes are recomputed as usual. *Fixed routes*: every block replays the originally selected routes. Drops are paired over images. These full-magnitude interventions differ in perturbation size from the matched-size decomposition in Appendix E.1.

Condition	Top-1 (%)	Drop, free routes (pp)	Drop, fixed routes (pp)
Original gain	56.054	–	–
Gain $\rightarrow$ spatial mean	34.472	21.58	20.46
Gain vectors shuffled across tokens ($\times 5$)	39.87–40.28	15.78–16.18	15.32–15.60

Predictions depend on how gain is placed. Table 3 intervenes on the gain at the point where it multiplies the retrieved signal response. Replacing the gain of all eight blocks by its within-image spatial mean lowers ImageNet top-1 accuracy from 56.05% to 34.47%, a paired drop of 21.58 points, and moving whole gain vectors to other tokens lowers it by 15.78–16.18 points. The drops remain almost unchanged when every block is forced to reuse its original routes (20.46 and 15.32–15.60 points); fixing the routes keeps the selected codes but still lets the perturbed gain change downstream activations, so the drops reflect the gain perturbation itself rather than a change of route: the prediction depends on which gain each token receives. The output assembly normalizes each token jointly over response factors and channels (Eq. 10), attenuating a common rescaling of its gains while retaining sensitivity to their relative allocation. The matched-size decomposition below supports this account at the tested checkpoint: the within-token pattern intervention has a larger accuracy effect than the level-only intervention, although level changes still matter. To test how the trained model uses gain, we decomposed each token's log gain into its mean level and its relative pattern across response slots and applied four interventions at a shared, label-free calibrated perturbation size. Across all eight blocks, flattening the within-token pattern reduced ImageNet top-1 by 2.36 pp (95% CI [2.12, 2.60]), compared with 1.86 pp [1.67, 2.07] for slot shuffling, 1.47 pp [1.27, 1.69] for a matched-size across-token move, and 0.65 pp [0.49, 0.81] for changing only the token-level gain. The flattening drop is about 3.6 times the level-only drop, supporting a stronger dependence on within-token gain allocation at this checkpoint. The approximately 2-pp drops use a deliberately small common perturbation, whereas the 21.58-pp drop uses the full spatial-mean intervention. Across individual blocks, flattening exceeds level-only in point estimate in all eight and peaks in blocks 2–3; block 6 responds more to slot shuffling, block 8 has no clear effect, and fixed-route replay yields a similar pattern (Appendix E.1).

What the gain and the codes carry across depth. We then measure, on COCO (5,000 val2017 queries against 20,000 train2017 gallery images), how the released response and the selected codes are organized over depth; all intervals are 95% image-level bootstrap intervals. First, the released response magnitude stays close to spatially uniform in every block (normalized spatial entropy 0.91–0.97), with a small decrease from shallow to deep blocks of 0.0041 [0.0040, 0.0043]; on the same 512 images, the trained model gives 0.0039 and three randomly initialized models give essentially zero. This spatial pattern is consistent with the output assembly and the matched-size gain decomposition: common gain scaling is attenuated by per-token normalization, while predictions at this checkpoint are more sensitive to changes in within-token gain allocation. Second, the selected codes form a largely shared set across classes with a consistent class-dependent shift: the same-minus-different-class affinity of named-slot code distributions is positive in every block and grows from 0.013 in the first block to 0.020 in the last, a shallow-to-deep increase of 0.0078 [0.0075, 0.0082]. On a separate exploratory COCO subset (256 selected queries, 251 valid queries, and 256 gallery images), the trained checkpoint's deep-minus-shallow Bhattacharyya-affinity contrast was +0.0104 (95% query-bootstrap CI [+0.0088, +0.0121]), reversing the negative contrasts of three random initializations (-0.0063 to -0.0053); this training-associated directional difference was measured with a selection rule distinct from the main COCO analysis. Most codes are reused across classes, and class identity is expressed as a small, depth-increasing change in how often each code is selected. Third, the two signals meet in the deep blocks: tokens in the top strength tercile show a larger class-related code affinity (0.028) than tokens in the bottom tercile (0.020), a difference of 0.0086 [0.0083, 0.0088]. Where the gain releases more response, the selected codes are more consistent within a class. An

exploratory gain-profile readout shows class affinity peaking at block 7 and receding at block 8, with retrieval mAP close to route histograms and below block representations (Appendix E.3).

4.5 Evidence Summary

The compute-matched ImageNet registry measures predictive accuracy and compactness, and the ImageNet-100 ablation compares composite response operators. At the frozen epoch-610 checkpoint, gain interventions and matched-size decomposition show prediction sensitivity to spatial placement and within-token allocation, including with original routes replayed. COCO code-affinity, random-initialization controls, and gain-profile readouts describe class-related structure across depth. CIFAR results provide smaller-scale context. Appendix D gives the protocols, Appendix E the extended diagnostics, and Appendix G the negative gain-cache result and other limitations.

5 Discussion

The epoch-610 interventions show that predictions depend on the spatial placement of the response gain, even when the original routes are replayed. At a matched perturbation size, changing the allocation among a token's response slots has a larger effect than changing only its common gain level. Released-response magnitude stays near-uniform across spatial tokens, while named-slot code distributions show a small class-related affinity that grows with depth and is larger in the high-strength group. On a separately selected exploratory COCO subset, the deep-minus-shallow affinity contrast is positive for the trained checkpoint and negative for three random initializations. This comparison measures directional differences in aggregate code affinity between the trained checkpoint and random initializations.

The biological analogy serves as a structural blueprint for functional decoupling—separating selective interaction, response retrieval, and state-dependent modulation—while the empirical claims rest strictly on the quantifiable machine learning behavior of the routed representations and their gains.

The compute-matched ImageNet registry uses a shared evaluation pipeline with training recipes that differ across model families; the ImageNet-100 ablation compares composite operators whose components change together.

6 Conclusion

Transmitter separates receptor-book selection, signal-book response retrieval, and input-dependent gain inside a vision backbone. Epoch-610 gain interventions show that this exposed response path affects predictions under the tested perturbations.

AI use statement

We used generative AI tools only to assist with editing and restructuring manuscript text, searching and summarizing related literature, drafting and reviewing analysis scripts, and critiquing experimental designs. Generative AI tools were not used to generate research ideas, model architectures, experimental designs, results, or data. All literature pointers were checked against the original sources, and all AI-assisted code and text were reviewed by the authors. We take responsibility for the final content of this work, including all text, claims, analyses, and artifacts produced with the aid of generative AI.

Ethics statement

This work is foundational vision research. It does not involve human subjects, private data, generative media, or deployment in decision-making systems, and it uses only standard public benchmark datasets under their terms of use (Appendix F.1). If the architecture were adapted to deployed vision systems, the usual concerns around robustness, fairness, privacy, and misuse of visual classifiers would apply.

Reproducibility statement

Section 4 defines the evaluated datasets, the ImageNet registry, and the fixed-checkpoint intervention protocol. Appendix D gives architecture settings, training recipes, and preprocessing; Appendix F

and Appendix F.1 give checkpoint provenance, compute metadata, and the asset inventory; and Appendix G states the scope limits of the evidence. The supplementary material contains an anonymized code package with the model implementation, training and evaluation scripts, command templates, environment notes, and dataset-layout instructions. Standard benchmark datasets are not redistributed.

REFERENCES

L. F. Abbott and Wade G. Regehr. Synaptic computation. *Nature*, 431(7010):796–803, 2004. doi: 10.1038/nature03010.

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebron, and Sumit Sanghai. GQA: Training generalized multi-query transformer models from multi-head checkpoints. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 4800–4810, 2023.

Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, and Simon Lacoste-Julien. A closer look at memorization in deep networks. In *International Conference on Machine Learning (ICML)*, pp. 233–242. PMLR, 2017.

Yoshua Bengio, Nicholas L'eonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

Vincent-Pierre Berges, Barlas Oguz, Daniel Haziza, Wen-tau Yih, Luke Zettlemoyer, and Gargi Ghosh. Memory layers at scale. In *International Conference on Machine Learning (ICML)*, pp. 3831–3842. PMLR, 2025.

Darshil Doshi, Aritra Das, Tianyu He, and Andrey Gromov. To grok or not to grok: Disentangling generalization and memorization on corrupted algorithmic datasets. In *International Conference on Learning Representations (ICLR)*, 2024.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021.

William Fedus, Barret Zoph, and Noam Shazeer. Switch Transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.

Dan Hendrycks and Thomas G. Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations (ICLR)*, 2019.

Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15262–15271, 2021.

Dimitris Kalimeris, Gal Kaplun, Preetum Nakkiran, Benjamin Edelman, Tristan Yang, Boaz Barak, and Haofeng Zhang. SGD on neural networks learns functions of increasing complexity. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, pp. 3491–3501, 2019.

Guillaume Lample, Alexandre Sablayrolles, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. Large memory layers with product keys. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, pp. 8548–8559, 2019.

Alexander H. Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. Key-value memory networks for directly reading documents. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1400–1409. Association for Computational Linguistics, 2016. doi: 10.18653/v1/D16-1147.

Bowen Pan, Rameswar Panda, Yifan Jiang, Zhangyang Wang, Rogerio Feris, and Aude Oliva. IA-RED^2: Interpretability-aware redundancy reduction for vision transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pp. 24898–24911, 2021.

Hubert Ramsauer, Bernhard Schäfl, Johannes Lehner, Philipp Seidl, Michael Widrich, Thomas Adler, Lukas Gruber, Markus Holzleitner, Milena Pavlović, Geir Kjetil Sandve, Victor Greiff, David Kreil, Michael Kopp, Günter Klambauer, Johannes Brandstetter, and Sepp Hochreiter. Hopfield networks is all you need. In *International Conference on Learning Representations (ICLR)*, 2021.

Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. DynamicViT: Efficient vision transformers with dynamic token sparsification. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pp. 13937–13949, 2021.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do ImageNet classifiers generalize to ImageNet? In *International Conference on Machine Learning (ICML)*, pp. 5389–5400. PMLR, 2019.

Carlos Riquelme, Joan Puigcerver, Basil Mustafa, Maxim Neumann, Rodolphe Jenatton, André Susano Pinto, Daniel Keysers, and Neil Houlsby. Scaling vision with sparse mixture of experts. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pp. 8583–8595, 2021.

Silvio O. Rizzoli and William J. Betz. Synaptic vesicle pools. *Nature Reviews Neuroscience*, 6(1):57–69, 2005. doi: 10.1038/nrn1583.

Peggy Sériès, Peter E. Latham, and Alexandre Pouget. Tuning curve sharpening for orientation selectivity: coding efficiency and the impact of correlations. *Nature Neuroscience*, 7(10):1129–1135, 2004. doi: 10.1038/nn1321.

Noam Shazeer. Fast Transformer decoding: One write-head is all you need. *arXiv preprint arXiv:1911.02150*, 2019.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations (ICLR)*, 2017.

Morgan Sheng and Myung Jong Kim. Postsynaptic signaling and plasticity mechanisms. *Science*, 298(5594): 776–780, 2002. doi: 10.1126/science.1075333.

Sarath Sivaprasad, Ankur Singh, Naresh Manwani, and Vineet Gandhi. The curious case of convex neural networks. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*, volume 12975 of *Lecture Notes in Computer Science*, pp. 531–546. Springer, 2021.

Sainbayar Sukhbaatar, Edouard Grave, Guillaume Lample, Hervé Jégou, and Armand Joulin. Augmenting self-attention with persistent memory. *arXiv preprint arXiv:1907.01470*, 2019.

Frederik Träuble, Anirudh Goyal, Nasim Rahaman, Michael C. Mozer, Kenji Kawaguchi, Yoshua Bengio, and Bernhard Schölkopf. Discrete key-value bottleneck. In *International Conference on Machine Learning (ICML)*, pp. 34431–34455. PMLR, 2023.

Misha V. Tsodyks and Henry Markram. The neural code between neocortical pyramidal neurons depends on neurotransmitter release probability. *Proceedings of the National Academy of Sciences of the United States of America*, 94(2):719–723, 1997. doi: 10.1073/pnas.94.2.719.

Aäron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, pp. 6306–6315, 2017.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, pp. 5998–6008, 2017.

Lu-Yang Wang and Leonard K. Kaczmarek. High-frequency firing helps replenish the readily releasable pool of synaptic vesicles. *Nature*, 394(6691):384–388, 1998. doi: 10.1038/28645.

Ross Wightman. PyTorch image models. `https://github.com/rwightman/pytorch-image-models`, 2019. GitHub repository.

Lei Zhu, Xinjiang Wang, Zhanghan Ke, Wayne Zhang, and Rynson W. H. Lau. BiFormer: Vision transformer with bi-level routing attention. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10323–10333, 2023.

Robert S. Zucker and Wade G. Regehr. Short-term synaptic plasticity. *Annual Review of Physiology*, 64(1): 355–405, 2002. doi: 10.1146/annurev.physiol.64.092501.114547.

A Appendix Overview

The appendix develops the route-object formulation and method details, records ImageNet registry and ImageNet-100 ablation protocols, provides CIFAR context, and presents the frozen epoch-610 gain interventions, spatial dispersion, named-slot code affinity, random-initialization control, and exploratory gain-profile analysis. Reproducibility notes, assets, and the negative gain-cache result complete the appendix.

Table 4 records how the main-text claims map to appendix material. The table separates the role of each appendix block so the main text can stay focused while the supporting protocols and diagnostics remain traceable.

Table 4: **Main-text to appendix map.** The appendix provides protocol and diagnostic detail for the evidence used in the main paper.

Main-text object	Appendix support	Evidence role
Analyses and supporting protocols		
Full-magnitude gain interventions	Section D.7	Epoch-610 placement and fixed-route protocol
Matched-size gain decomposition	Section E.1	Epoch-610 within-token allocation
Spatial dispersion and code affinity	Section E.2	Epoch-610 definitions and evidence boundary
Exploratory gain profile	Section E.3	Separate epoch-610 readout
Route and gain notation	Section B	Routed-event coordinates and internal row reuse
Training and evaluation recipes	Section D.1	Checkpoint provenance and core settings
ImageNet registry and compactness	Sections D.2–D.3	Shared evaluation pipeline
ImageNet-100 ablation	Section D.6	Composite operator comparison
CIFAR context and training dynamics	Sections D.4–D.5	Smaller-scale context
Negative gain-cache result	Section G	Epoch-610 few-shot fusion result
Biological analogy and assets	Sections B.2, F.1	Design boundary and inventory

B Route-Object Formulation

The main paper describes Transduction as a route-object formulation. A route object is an exposed forward-time state consisting of a selected receptor-book entry, a separately retrieved signal-book response, and an input-dependent modulation value. The state is computed from the model's route, response, and gain operations during the standard forward pass, without an added supervision target or a post-hoc unit label. It is produced by the normal forward pass and can therefore be logged, counted, or perturbed. Reuse denotes repeated selection of a learned receptor/signal row across events.

This separates Transduction from several nearby mechanisms. Standard self-attention exposes attention weights, but the value-side response is a continuous mixture rather than a persistent learned route entry with a separate signal-book response. Standard vector quantization exposes a discrete assignment, but the matching code and emitted representation are typically coupled. Hard key-value retrieval exposes a key and a value table, but does not by itself provide the full response-expression path used here. Mixture of experts routes tokens to coarse modules; Transduction instead records fine-grained route choices inside the visual backbone.

The point is more specific: Transduction makes a particular internal object visible–the query-conditioned route between matching and response expression–and the experiments test whether that object is useful enough to keep.

For a routed event i, σ_i is the selected strength logit, $\gamma_i = \mathrm{sigmoid}(\mathrm{clip}(\sigma_i, -20, 20) - \pi_{b,k_i^\star})$ is its gate, and $g_i = \mathrm{softplus}(a_{k_i^\star})\gamma_i$ is the full gain that multiplies the retrieved response $\bar{v}_i$; $r_i = g_i \bar{v}_i$ is the released response. A named response slot is a fixed factor–chunk coordinate inside a block and head. Thus a token's gain vector spans its named slots, while a code histogram pools selected route IDs at a specified slot across spatial tokens. The gain-profile subsection uses ρ_g for a centered log-*gate* profile and ρ_u for a centered log-*full-gain* profile; neither is the selected strength σ_i.

B.1 Detailed Transduction Matching Diagram

Figure 2B gives the detailed matching view after the multi-head overview in Figure 2A. The diagram shows query fragments, receptor and signal books, modulation, and routed outputs as engineered computational components. It expands the route object described in Section B: a selected receptor-book entry, a separately retrieved signal-book response, and an input-dependent modulation value.

B.2 Biological Inspiration Boundary

The motivation is a role-level analogy: biological signaling suggests that identity, selectivity, amplitude, and recent activity can jointly shape an effective response [B1–B4]. Transduction uses this only as inspiration for separating *what is selected*, *what response is expressed*, and *how strongly that response is modulated*.

Transduction turns this decomposition into an ML operator. A query subvector selects a discrete route, the selected route retrieves a separately learned signal-book response, and a modulation term controls the expressed response. The organization of signal transmission inspires the operator's separation of selection, response retrieval, and modulation. Transduction is an engineered neural-network operator whose components perform discrete selection, learned response retrieval, and input-dependent modulation.

Figure 3 illustrates the role separation in biological signaling that motivates the operator. It illustrates that biological signal transmission suggests a role separation among identity, amplitude, and modulation; the proposed architecture implements an engineered version of that role separation for visual features.

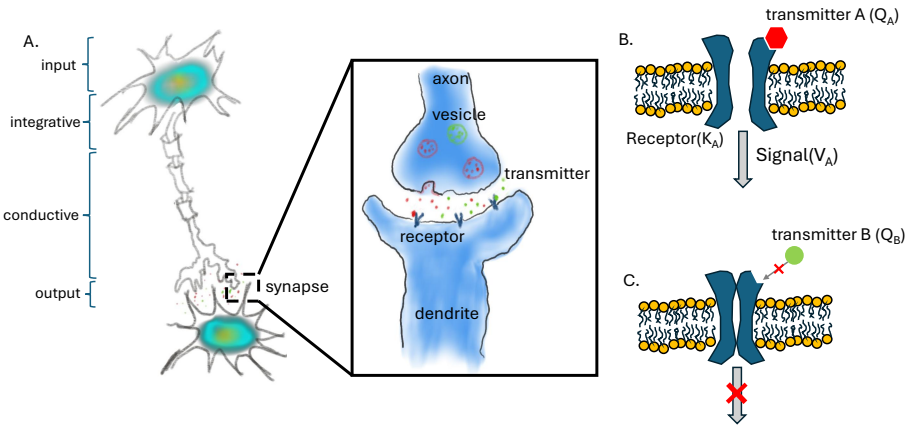


Figure 3: **Role-level biological inspiration.** The figure records the design analogy behind the terminology: selective interaction, response expression, and state-dependent modulation.

B.3 Biological Role Translation

The intended analogy is summarized in Table 5. The table relates biological signaling concepts to computational roles in Transduction. This distinction matters. A receptor is a cellular object, while a learned codebook row is a model parameter. However, both can play a selective role in a larger signaling process. Release probability and learned strength provide a role-level analogy for the magnitude of response expression in biological signaling and the operator, respectively.

Several biological ideas are especially relevant as inspiration. Classical work on quantal transmitter release emphasizes that communication can be organized around discrete release events rather than a purely continuous wire-like signal [B5–B7]. Experiments on cortical synapses further show that

Table 5: **Biology-inspired role translation.** The analogy is used to motivate a decomposition of computation into selection, response expression, and modulation.

Biological intuition	Computational role	Transduction design object	Boundary
Receptor selectivity	Discrete state selection	Receptor book / active route index	Discrete route selection
Transmitter/receptor pairing	Response identity	Separate signal book selected by route	Learned response lookup
Release probability / amplitude	Response strength	Selected strength and modulation value	Learned response modulation
Short-term usage effects	State-dependent regulation	Fatigue-style activity gate	Usage-dependent response gating
Repeated pathway use	Reusable response basis	Logged route/code traces across inputs	Repeated codebook-row selection

changes in synaptic efficacy and release probability can change downstream response statistics rather than acting only as a scalar gain [B8,B9]. Postsynaptic signaling studies further show that receptor composition and trafficking help determine which downstream response machinery is engaged, making response identity separable from raw signal amplitude in a useful role-level sense [B10,B11]. Work on short-term synaptic plasticity and use-dependent feedback shows that recent activity can modulate effective transmission over short timescales, sometimes changing information transfer rather than merely scaling it [B12–B16]. Sensory signal transduction also illustrates that selectivity, amplification, and adaptation can be intertwined in a compact signal-processing interface rather than placed only at the final readout [B17]. These references motivate the separation of route identity, response retrieval, and modulation in the vision operator.

The resulting ML hypothesis is that explicit selected states in the query-to-response computation improve backbone inspectability. The epoch-610 experiments test gain placement and trace structure.

C Method Details

The main method section keeps the core equations and architecture compact. This section provides implementation-facing detail for the two parts most likely to be checked by a reviewer: the fatigue-style modulation and the training loss used to make the receptor book trainable.

C.1 Usage-Dependent Gain

The fatigue-style module is an engineered activity-dependent gate. It implements a differentiable modulation rule motivated by the finite-resource analogy for repeated state selection. The selected signal-book response is multiplied by a scalar derived from route usage and a learned descriptor. This provides a bounded input-dependent expression strength for each selected response.

Using the notation from Section 3, the released code implements this as $g_i = \mathrm{softplus}(a_{k_i^\star})\,\mathrm{sigmoid}(\sigma_i - \log f_{\mathrm{use}}(\pi_{b,k_i^\star}))$, with $f_{\mathrm{use}}(\pi) = \exp(\pi)$. This parameterization implements the response-expression gate.

Algorithm 1 Fatigue-style usage-dependent response gate used in Transmitter

Require: `act_hot`, `strength`, `vesicle_amt`, selected signal response $\bar{v}_{b,e}$
1: `route_count`: $n_{b,k} \leftarrow \sum_e \texttt{act_hot}_{b,e,k}$
2: `usage_ratio`: $\pi_{b,k} \leftarrow n_{b,k} / \sum_j n_{b,j}$
3: `active_route`: $k^\star(b,e) \leftarrow \arg\max_k \texttt{act_hot}_{b,e,k}$
4: `usage_coeff`: $u_{b,e} \leftarrow \exp(\pi_{b,k^\star(b,e)})$
5: `amplitude`: $A_{b,e} \leftarrow \mathrm{softplus}(\texttt{vesicle_amt}_{k^\star(b,e)})$
6: `gate`: $g_{b,e} \leftarrow A_{b,e}\,\mathrm{sigmoid}(\texttt{strength}_{b,e} - \log u_{b,e})$
Ensure: `released_response`: $g_{b,e}\bar{v}_{b,e}$

C.2 Synapse-Style Codebook Objective

The receptor book is selected through nearest-neighbor assignment, so the training objective must provide useful gradients for both the query fragments and the codebook entries. The objective combines a self-distillation-style representation term with an affinity term between query fragments and selected matching entries. A usage-ratio factor discourages degenerate allocation in which only a

tiny subset of codes carries all assignments. This auxiliary term regulates codebook allocation during training.

The hard nearest-neighbor index is not itself relaxed into a soft probability distribution in the forward computation. Consequently, gradients from the classification loss update the modules on the selected response path, while query-side matching geometry and key entries are shaped by the auxiliary codebook objective and its stop-gradient terms. This surrogate training scheme updates hard-routing geometry through the auxiliary codebook objective.

Algorithm 2 Synapse-style auxiliary objective used for receptor-book training

Require: `q`, `active_keys`, `active_indices_onehot`
1: `counts`: $n_{b,k} \leftarrow \sum_e$ `active_indices_onehot`$_{b,e,k}$
2: `sample_prob`: $p_{b,k} \leftarrow n_{b,k} / \sum_j n_{b,j}$
3: `batch_prob`: $p_{\mathcal{B},k} \leftarrow \sum_b n_{b,k} / \sum_{b,j} n_{b,j}$
4: `usage_reward`: $r_b \leftarrow \mathrm{stopgrad}(H(p_{\mathcal{B}}) - H(p_b))$
5: `concept_loss`: align routed `q` features with a stopped and centered projection target
6: `commit_loss`, `vq_loss`: stop-gradient distances between `q` and `active_keys`
7: `commit_total`: usage-weighted combination of `commit_loss` and `vq_loss`
8: `syna_loss`: (`commit_total` + `concept_loss`) $\cdot$ `commit_reg`
Ensure: auxiliary receptor-book loss `syna_loss`

The hyperparameter setting used in the reported experiments is $\lambda_1 = 2.5$, $\lambda_2 = 1$, $\lambda_3 = 2.5$, $\lambda_4 = 1$, and $\lambda_5 = 10^{-4}$. Stop-gradient is used to separate the query-side and codebook-side updates in the representation and affinity terms. The loss trains the codebook through representation and affinity terms. Algorithms 1 and 2 are schematic summaries; tensor reshaping, clipping, running-center updates, and numerical constants are left to the released implementation.

D Experiment Protocol Cards

This section records protocols for the registry, operator ablation, and epoch-610 interventions, together with CIFAR context and training diagnostics. Section F identifies the checkpoint source and supporting metadata.

D.1 Reproducibility Recipe Cards

Table 6 records the retained core settings for the reported training and fixed-checkpoint evaluations. ImageNet-1K and ImageNet-100 use the same base recipe family—the same optimizer family, epoch budget, total batch size, and augmentation family—with only the reduced-scale ImageNet-100 adjustments listed explicitly in the table. The ImageNet registry trains its evaluated models from scratch, instantiating non-Transmitter baselines from standard `timm` (Wightman, 2019) architecture definitions.

The ImageNet-100 operator ablation compares final-validation accuracy; the fixed-checkpoint diagnostics measure gain interventions and aggregate route statistics at epoch 610.

D.2 ImageNet Registry

The ImageNet registry records the evaluated entries under the protocols used by the main paper. Table 1 tracks parameter count, FLOPs, and ImageNet-1K accuracy. Separate registry evidence records ImageNet-V2 accuracy, ImageNet-A accuracy under the 1000-class protocol, and ImageNet-C mCE. ImageNet-C mCE follows the Hendrycks-Dietterich convention and is normalized by the AlexNet top-1 corruption-error reference over 15 corruptions and five severities. The main table therefore calibrates ImageNet-scale performance, parameter count, and FLOPs for the registry.

Transmitter-768 is strong in the reported registry setting, especially relative to its parameter count, and the table is intended to compare the evaluated rows under the recorded training and evaluation choices.

Table 6: **Condensed recipe cards.** These cards keep only the core training and rerun settings needed for the main experiment classes.

Experiment class	Core settings
ImageNet Pretraining (1K & 100)	• **Data:** ImageNet-1K or CMC ImageNet-100 subset (seed 0), at 224-pixel resolution. • **Training schedule:** total batch size 1024 (4×256), 1000 epochs, checkpoints every 50 epochs. Warmup iterations: 5000 for IN-1K, 500 for IN-100. • **Optimization & Objective:** AdamW, learning rate 10^{-4}, weight decay 10^{-4} (none on codebook parameters). Cross-entropy plus Synapse-style auxiliary objective. • **Augmentation / eval preprocessing:** random resized crop (scale 0.3–1.0), horizontal flip, 15° rotation, color jitter, fixed normalization, and deterministic validation resizing. • **Architecture:** IN-1K uses full-scale Transmitter models; IN-100 ablations use 4 operator layers, code dim 256, 4 heads, drop-path rate 0.3.
Current epoch-610 analyses	• **Checkpoint:** frozen `checkpoint_epoch_0610.pt` from the eight-block run and its source snapshot. • **Modes:** full-magnitude gain interventions, matched-size decomposition, spatial dispersion, named-slot code affinity, and exploratory gain profiles. • **Data:** ImageNet-1K validation for interventions; COCO val2017 queries and train2017 gallery for trace diagnostics.

Table 7: **Cross-dataset baseline context.** CIFAR-10/100 final-validation (epoch-1000) top-1 accuracy, sorted by parameter count.

Model	Params (M) ↓	CIFAR-10 Top-1 ↑	CIFAR-100 Top-1 ↑
Transmitter-256	4.35	88.72	88.38
ViT-Tiny	5.72	85.13	55.88
ConvNeXtV2-Pico	9.07	90.49	64.36
ResNet-18	11.69	90.95	64.64
ViT-Small	22.05	83.91	53.90
ResNet-50	25.56	92.13	67.44
ConvNeXtV2-Tiny	28.64	92.22	63.65

The compactness pattern is clearest for Transmitter-768. It uses 10.221M parameters and 2.923G FLOPs, compared with 22.051M and 8.482G for ViT-Small, 25.557M and 8.178G for ResNet-50, and 28.635M and 8.911G for ConvNeXtV2-Tiny. Its final-validation top-1 is 63.094%, compared with 63.319% for ResNet-50. These values quantify learned-weight compactness and nominal compute for the evaluated architectures.

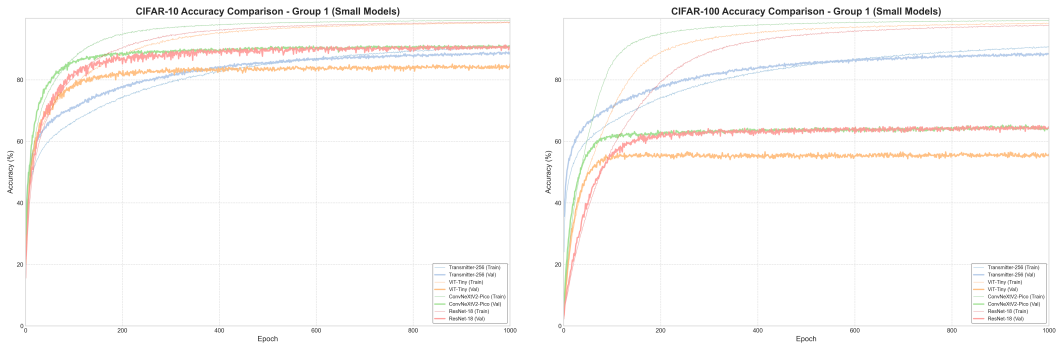
Reported FLOPs represent whole-model totals computed using PyTorch `FlopCounterMode`. The current Transduction implementation performs dense matching between each query fragment and all K receptor entries. For batch size N, L spatial tokens, S signal factors, J routed chunks, and receptor dimension r, this matching step scales as $\mathcal{O}(NLSJKr)$. This records the audited computation used for the nominal FLOP count, while the parameter-count result captures the evaluated backbone's compact weight footprint.

D.3 Compactness

The compactness claim in this paper is a learned-weight and nominal-compute claim. Parameter counts measure the size of the learned backbone, and the FLOP counts above record the audited PyTorch counter estimates used for the registry. These indicators are the resource evidence used by the paper: Transmitter-768 has a smaller parameter budget and lower nominal FLOPs than the larger external baselines, while its final-validation accuracy is close to that of ResNet-50.

D.4 Cross-Dataset Context

The CIFAR-10/100 comparison is kept as appendix context because it uses smaller-scale benchmarks. The main paper presents its ImageNet-scale comparison in Table 1.



(a) CIFAR-10 small models.

(b) CIFAR-100 small models.

Figure 4: **CIFAR-10/100 small-model paired training curves.** Raw training-log views for the small-model groups summarized in Table 8. Each panel shows logged training and validation accuracy for one dataset. These curves describe recorded training and validation trajectories across epochs.

Table 8: **Matched train/validation logging gaps.** This table is a training-log diagnostic for the persistence of validation-minus-training accuracy inversions over 1000 paired epochs per row. Positive-gap epochs count how often logged validation accuracy exceeds logged training accuracy; longest run is the longest consecutive positive-gap interval.

Model	Positive-gap epochs	Longest run	Mean gap (pp)
CIFAR-10			
Transmitter-256	566	503	+0.98
ResNet-18	118	105	-5.79
ConvNeXtV2-Pico	83	77	-6.75
ViT-Tiny	81	29	-10.32
ResNet-50	72	28	-5.40
ConvNeXtV2-Tiny	55	54	-6.75
ViT-Small	45	40	-13.29
CIFAR-100			
Transmitter-256	570	534	+1.06
ResNet-18	80	75	-25.24
ResNet-50	49	44	-26.98
ConvNeXtV2-Pico	34	33	-31.25
ConvNeXtV2-Tiny	25	24	-32.97
ViT-Tiny	41	36	-35.70
ViT-Small	20	18	-41.13

D.5 Training-Dynamics Diagnostic

In addition to endpoint accuracy, the paired CIFAR logs expose a training dynamic that we report as a diagnostic. Figure 4 provides train/validation accuracy curves for the CIFAR-10/100 small-model groups. Table 8 summarizes the retained paired logs. In the CIFAR-10 Transmitter-256-BYOL run, validation accuracy is higher than logged training accuracy for 566 of 1000 epochs, with a consecutive positive-gap interval spanning epochs 1–503.

For the resumed CIFAR-100 Transmitter logs, the diagnostic stitches predecessor and resumed train/validation metric histories by epoch step, keeping the later timestamp in overlapping steps.

This diagnostic characterizes empirical training dynamics within the evaluated runs. A prior study of input-output convex neural networks reports negative generalization gaps, self-regularization, and resistance to memorization (Sivaprasad et al., 2021). Separately, memorization studies show that deep networks often learn simple or shared patterns before fitting idiosyncratic examples, and that regularization can suppress memorization while preserving performance on real data (Arpit et al., 2017); related SGD dynamics work frames training as a progression from simpler to more complex functions (Kalimeris et al., 2019). In corrupted-label algorithmic settings, regularization can even make models ignore corrupted training labels while maintaining high accuracy on uncorrupted examples (Doshi et al., 2024). These works provide context for train/validation inversions in the

input-output convex-network, memorization, SGD, and corrupted-label settings described above; the Transmitter observation is measured in the paired CIFAR training logs.

These trajectories use the training and validation accuracies recorded under the runs' augmentation, stochastic-layer, and evaluation settings.

D.6 Reduced-Scale Operator Ablation

The ImageNet-100 controlled operator ablation asks whether the full routing-expression operator is useful relative to simpler alternatives when the surrounding system is held fixed. The controlled variables are the reduced ImageNet split, surrounding encoder, resolution, optimizer, and training recipe. The changed variable is the routing operator.

The reported ablation rows use the recipe card in Table 6: ImageNet-100 is implemented as the 100-class CMC WNID subset of ImageNet-1K and is launched through the shared ImageNet-100 operator-ablation artifact with seeds 0–2, 4 layers, batch size 256, 1000 epochs, the augmentation/evaluation transforms listed in the recipe card, and the same optimizer-family defaults. Run metadata records the split size and execution parameters of the operator comparison, which uses a fixed execution protocol and data split.

The evaluated operator families are:

- **Full Transduction**: hard nearest-key assignment, separate response retrieval, and modulated response expression.
- **Soft-KV plain**: soft key weights and a weighted value mixture.
- **Strict VQ**: hard nearest-code assignment where the selected code itself is emitted as the representation.
- **Hard-KV plain**: hard nearest-key lookup without the full response-expression path.
- **Hard-KV detached-surrogate diagnostic**: a detached component diagnostic run only as an implementation check; it is not part of the compact comparison table.

The variant with the highest final accuracy in this reduced-scale study keeps matching and response expression separated while preserving the response-expression path. Overall, the table provides empirical support for the evaluated operator configuration within the fixed protocol.

D.7 Epoch-610 Full-Magnitude Gain Intervention Protocol

At the response-multiplication point of Eq. 9, the current experiment replaces the full scalar gain g_i for every routed event in all eight blocks. One arm substitutes the within-image spatial mean at the same named slot; the other moves each token's whole vector of named-slot gains to another spatial token, with five permutations. Thus slot coordinates remain together in the second arm. Free-route inference recomputes downstream routes; fixed-route replay forces the baseline selected indices in every block while allowing activations to change. Table 3 gives the ImageNet-1K validation results, including the five-shuffle range. The paired accuracy drops use images as resampling units where a confidence interval is reported. Downstream activation changes are part of the intervention response, including during fixed-route replay.

E Extended Results

E.1 Matched-Size Gain Decomposition and Normalization Diagnostics

Using one frozen epoch-610 checkpoint, we intervened on the full gain $u_{t,m}$ of Eq. 9 before response multiplication and the per-token output normalization of Eq. 10. For token t and named response slot m, write $y_{t,m} = \log u_{t,m} = \alpha_t + \rho_{t,m}$, where α_t is the mean log gain over the slots and $\rho_{t,m} = y_{t,m} - \alpha_t$. The four arms attenuate ρ toward zero (flatten pattern), permute ρ among a token's slots (slot shuffle), move the gain vector across tokens (across-token move), or permute α across tokens while retaining ρ (level only).

Table 9: **Matched-size gain interventions at epoch 610.** ImageNet-1K validation (50,000 images), free routes. Entries are paired top-1 drops in pp [95% image-level bootstrap CI], averaged over five intervention seeds per image. Bold marks the largest point estimate in a row, except block 8, where all intervals include zero.

Block	Flatten pattern	Shuffle slots	Move across tokens	Level only
1	**0.44** [0.28, 0.61]	0.26 [0.13, 0.39]	0.14 [0.01, 0.27]	0.11 [−0.00, 0.22]
2	**0.61** [0.45, 0.78]	0.54 [0.41, 0.68]	−0.01 [−0.14, 0.11]	0.09 [−0.01, 0.21]
3	**0.62** [0.44, 0.79]	0.37 [0.24, 0.50]	0.17 [0.04, 0.29]	0.20 [0.07, 0.33]
4	**0.23** [0.09, 0.35]	0.17 [0.06, 0.27]	0.07 [−0.02, 0.16]	0.02 [−0.05, 0.09]
5	**0.36** [0.21, 0.50]	0.27 [0.15, 0.38]	0.22 [0.10, 0.34]	0.02 [−0.06, 0.09]
6	0.17 [0.06, 0.28]	**0.27** [0.16, 0.37]	0.10 [0.00, 0.20]	−0.01 [−0.08, 0.06]
7	**0.27** [0.15, 0.38]	0.20 [0.10, 0.29]	0.14 [0.06, 0.23]	0.12 [0.06, 0.19]
8	0.08 [−0.02, 0.18]	0.08 [−0.01, 0.17]	0.09 [−0.01, 0.18]	0.03 [−0.02, 0.07]
All blocks	**2.36** [2.12, 2.60]	1.86 [1.67, 2.07]	1.47 [1.27, 1.69]	0.65 [0.49, 0.81]

Table 10: **Epoch-610 block summaries.** Normalized spatial entropy of released-response RMS and same-minus-different-class named-slot code affinity. Values are descriptive block aggregates.

Block	1	2	3	4	5	6	7	8
Spatial entropy	.931	.972	.965	.935	.947	.906	.936	.959
Code affinity	.0132	.0112	.0085	.0101	.0166	.0184	.0197	.0203

All four arms use a shared log-gain perturbation size $\delta = 0.9956$. Using baseline u on 128 ImageNet training calibration images, without labels or logits, candidate distances d were computed for nine scopes (all blocks and each of eight single blocks); δ was set to half the smallest d. At this calibrated perturbation scale, the matched arms produce approximately 2-pp drops, reported separately from the 21.58-pp full spatial-mean intervention in Table 3, which uses a different perturbation size.

Table 9 reports paired top-1 drops on the 50,000 ImageNet validation images, averaging five intervention seeds per image; brackets are 95% image-level bootstrap intervals. Flattening has a larger point estimate than level only in every block, with the largest single-block flattening drops in blocks 2–3. Slot shuffling instead has the largest point drop in block 6. None of the four block-8 intervals excludes zero; level-only replacement has intervals excluding zero in blocks 3 and 7. Replaying original routes changes each single-block drop by less than 0.1 pp and yields zero downstream route changes. The single-block effects are measured in separate interventions; the all-block effect is measured in a joint intervention.

Across the same 50,000 images and 32 block/head combinations, the ratio of post- to pre-sortingNorm token RMS varies from 2.8× to 529×. Several strongly amplified heads have pre-norm RMS of roughly 10^{-3} to 2×10^{-4}, near the scale of the 10^{-5} epsilon. The across-token Pearson correlation between pre- and post-norm RMS is near +0.9 in some heads but negative in others (e.g., -0.55 for block 6/head 1 and -0.25 for block 8/head 3). Across-token coefficients of variation decrease after normalization in most heads. These measurements support attenuation of common gain magnitude with substantial head-to-head variation.

E.2 Epoch-610 Spatial Dispersion and Code Affinity

The spatial quantity is the per-token RMS of the released response r_i, pooled over response factors and channels, then normalized across spatial tokens before spatial entropy is divided by the log token count. This measures the spatial dispersion of released-response RMS. For code affinity, each image has a code-frequency distribution for each named factor–chunk slot within a block and head. Same-class minus different-class Bhattacharyya affinity is computed with COCO val2017 queries as anchors and train2017 gallery images as partners; the reported block values aggregate over the named slots and heads. The high and low token groups are termed "strength terciles". Head-input norm and center-distance rankings gave larger deep-group contrasts than the strength-tercile ranking in this epoch-610 analysis.

The main COCO analysis used 5,000 queries and 20,000 gallery images. For code-affinity contrasts, the confidence intervals measure query-resampling variability with the gallery and trained checkpoint

Table 11: **Exploratory gain-profile class affinity at epoch 610.** (a) Same-minus-different-class cosine affinity Δ for ρ_g, averaged over four heads. (b) Class-macro retrieval mAP at block 8. Affinity magnitudes are compared within each representation, using its own dimension and distance definition.

(a) Gain-profile affinity across blocks

Block	1	2	3	4	5	6	7	8
ρ_g Δ	.0141	.0096	.0085	.0150	.0160	.0186	.0227	.0153

(b) Block-8 class-macro mAP

Representation	mAP	Representation	mAP	Representation	mAP
ρ_g	.413	ρ_u	.413	Route histogram	.411
Selected-code constant	.389	Raw u	.391	Head-input norm	.372
Centre distance	.371	Value norm	.379	Block C	.494
Block output	.502				

held fixed. The released-response shallow-minus-deep spatial entropy difference was 0.00415 [0.00402, 0.00429], and the code-affinity deep-minus-shallow difference was 0.00782 [0.00746, 0.00817]. The high-minus-low strength-tercile contrast in deep blocks was 0.00855 [0.00828, 0.00884].

In a separate exploratory COCO subset with 251 valid queries and 256 gallery images, the trained checkpoint's deep-minus-shallow code-affinity contrast was +0.0104, whereas three random initializations gave negative contrasts (-0.0063 to -0.0053). This direction reversal concerns the deep-minus-shallow contrast under that subset's selection rule; the same-minus-different affinity retains its sign, and the trained checkpoint is shared with the main analysis. The subset and main COCO analysis use different selection rules.

E.3 Gain-Profile Class Affinity Across Depth

We examine one frozen epoch-610 checkpoint in an exploratory analysis, using BF16 inference and FP32 statistics. Within a fixed block ℓ and head h, let b index the image, t a valid token, and m a named response slot. The gate factor in Eq. 9 is $\gamma_{t,m} = \mathrm{sigmoid}(\mathrm{clip}(\sigma_{t,m}, -20, 20) - \pi^{\ell,h}_{b,k_{t,m}})$; π is the image-level usage fraction of the selected route. We form the centered log-gate profile $\rho_g(t,m) = \log \gamma_{t,m} - M^{-1} \sum_{m'} \log \gamma_{t,m'}$ over M slots, and define ρ_u analogously from the full gain $u_{t,m} = \mathrm{softplus}(a_{k_{t,m}})\gamma_{t,m}$. Each image is represented by the mean profile over its valid tokens. On COCO, 5,000 val2017 queries rank 20,000 train2017 gallery images (4,952 valid queries). We measure same-class minus different-class cosine affinity Δ, averaged over four heads, and class-macro retrieval mAP.

The ρ_g affinity decreases from block 1 (.0141) to block 3 (.0085), rises through block 7 (.0227), and falls at block 8 (.0153). Its block-7–8 minus block-1–2 contrast is .00711 cosine (95% query-and-gallery bootstrap CI [.00656, .00768]; label-permutation null $p = 0.005$). Within identical-route groups, ρ_g shows positive class affinity (.0087, reaching .0164 at block 7 and .0144 at block 8); this is a route-conditioned measurement. At block 8, its .413 mAP matches ρ_u (.413) and is close to the route histogram (.411); the selected-code constant reaches .389. Block C (.494) and the block output (.502) give substantially stronger class-macro retrieval readouts. The gain profile thus exposes class-related structure in this checkpoint's internal trace, with a retrieval signal similar to the route histogram and below the block features.

F Implementation and Reproducibility Notes

The manuscript is organized so that the main paper carries the compact claim and the appendix carries rerun detail.

F.1 Asset and Compute Metadata

Table 12 records the existing datasets and software assets used by the paper, the source or version information available from the manuscript artifacts, and whether the asset is redistributed. The benchmark datasets are used from their original sources and are not packaged with the submission.

Table 12: **Existing asset and license/terms inventory.** The paper uses standard external datasets and software dependencies but does not redistribute restricted benchmark images. Source URLs are included where available; dataset users remain responsible for satisfying the terms at the original source.

Asset	Role in paper	Source / version	License or terms noted at source	Redistributed?
ImageNet-1K / ILSVRC2012	Training and evaluation for the ImageNet registry and route diagnostics	Official ImageNet access site, `https://image-net.org/download.php`	ImageNet access agreement; non-commercial research and educational use; ImageNet does not own image copyrights	No
ImageNet-V2 matched-frequency	Robustness/generalization evaluation	ImageNetV2 release, `https://imagenetv2.org/`; repository `https://github.com/modestyachts/ImageNetV2`	MIT license for repository code; image data follow source-image metadata/terms described by the release	No
ImageNet-A	Natural-adversarial evaluation under the 1000-class protocol	Natural Adversarial Examples release, `https://github.com/hendrycks/natural-adv-examples`	MIT license for repository code; image files follow release/source-image terms where applicable	No
ImageNet-C	Corruption robustness evaluation and mCE computation	ImageNet-C Zenodo release, `https://zenodo.org/records/2235448`; robustness repository `https://github.com/hendrycks/robustness`	Apache-2.0 license for the robustness release repository; dataset files obtained from the original release	No
CIFAR-10 / CIFAR-100	Appendix cross-dataset context and training-dynamics diagnostics	Official CIFAR page, `https://www.cs.toronto.edu/~kriz/cifar.html`	Official source distribution; no redistribution in this submission	No
`timm`	Baseline architecture definitions; no public pretrained weights used for the registry	Version 1.0.24 in the inference artifact; `https://pypi.org/project/timm/`	Apache-2.0	Dependency only
PyTorch	Training, evaluation, FLOP counting, and inference profiling framework	Version 2.10.0+`cu130` in the inference artifact; `https://pytorch.org/`	BSD-style / BSD-3-Clause license	Dependency only

Table 13: **Compute metadata and configuration specifications.** Retained compute metadata and diagnostic script parameters by experiment class.

Experiment class	Metadata / parameter specifications
ImageNet-scale Transmitter training	Batch size, epoch count, optimizer family, learning rate, weight decay, warmup, augmentation/preprocessing recipe, checkpoint provenance, and checkpoint interval are recorded in the appendix recipe cards and artifacts.
Compute-matched ImageNet baseline registry	Same-registry scope, architecture source, checkpoint provenance, evaluation datasets, metrics, parameter counts, FLOPs, and input shape are recorded.
ImageNet-100 controlled operator ablation	Seed, 100-class split, batch size, epoch count, layer/code/head settings, optimizer settings, and `torchrun` process count are recorded.
Fixed-checkpoint diagnostics	Diagnostic script parameters: checkpoint path, checkpoint-matched source, strict-load status, dataset root, batch size, seed, episode split, and output directory.
Nominal inference-resource indicators	Parameter counts, PyTorch `FlopCounterMode` estimates, input shape, registry scope, and the dense receptor-matching scaling note are recorded.

Table 13 outlines retained compute metadata and diagnostic script parameter specifications across experiment classes.

F.2 Source and Artifact Notes

The current gain, spatial-dispersion, and code-affinity analyses use the frozen eight-block `checkpoint_epoch_0610.pt` and its run's source snapshot.

G Additional Limitations

The evidence is scoped to route-explicit operator behavior in the reported training and evaluation settings. The main limitations are as follows.

- Negative gain-cache result: in a separate exploratory epoch-610 evaluation, equal-weight gain-cache fusion lowered feature-only few-shot top-1 accuracy in all six Flowers102, Pets, and DTD shot/dataset settings (paired changes from -0.17 to -2.88 percentage points).
- The ImageNet registry uses a shared evaluation pipeline and from-scratch training, with differing recipes and seed counts across model families.
- The hard routing path uses surrogate codebook training rather than a directly differentiable relaxation of the nearest-neighbor index.
- The ImageNet-100 composite ablation compares configurations that jointly change gain, the factorwise RBF mask, and response decoupling.
- Epoch-610 affinity is small and describes class-dependent shifts in aggregate code distributions.
- The biological framing motivates the engineered separation of selection, response retrieval, and modulation.

The current checkpoint evidence supports sensitivity to gain placement and within-token allocation, with descriptive spatial-dispersion and code-affinity diagnostics.

H Appendix References

[B1] Kandel, E. R. et al. *Principles of Neural Science*. McGraw-Hill, 2021.

[B2] Kennedy, M. B. Synaptic signaling in learning and memory. *Cold Spring Harbor Perspectives in Biology*, 2013.

[B3] Citri, A. and Malenka, R. C. Synaptic plasticity: multiple forms, functions, and mechanisms. *Neuropsychopharmacology*, 2007.

[B4] Abbott, L. F. and Regehr, W. G. Synaptic computation. *Nature*, 2004.

[B5] Katz, B. *The Release of Neural Transmitter Substances*. Liverpool University Press, 1969.

[B6] Augustine, G. J. and Kasai, H. Bernard Katz, quantal transmitter release and the foundations of presynaptic physiology. *The Journal of Physiology*, 2007.

[B7] Ribrault, C., Sekimoto, K., and Triller, A. From molecular stochasticity to variability of synaptic transmission. *Nature Reviews Neuroscience*, 2011.

[B8] Markram, H. and Tsodyks, M. Redistribution of synaptic efficacy between neocortical pyramidal neurons. *Nature*, 1996.

[B9] Tsodyks, M. V. and Markram, H. The neural code between neocortical pyramidal neurons depends on neurotransmitter release probability. *PNAS*, 1997.

[B10] Sheng, M. and Kim, M. J. Postsynaptic signaling and plasticity mechanisms. *Science*, 2002.

[B11] Shi, S. H. et al. Subunit-specific rules governing AMPA receptor trafficking to synapses in hippocampal pyramidal neurons. *Cell*, 2001.

[B12] Zucker, R. S. and Regehr, W. G. Short-term synaptic plasticity. *Annual Review of Physiology*, 2002.

[B13] Hennig, M. H. Theoretical models of synaptic short term plasticity. *Frontiers in Computational Neuroscience*, 2013.

[B14] Scanziani, M. et al. Use-dependent increases in glutamate concentration activate presynaptic metabotropic glutamate receptors. *Nature*, 1997.

[B15] Salmasi, M., Loebel, A., Glasauer, S., and Stemmler, M. Short-term synaptic depression can increase the rate of information transfer at a release site. *PLOS Computational Biology*, 2019.

[B16] Lou, X. et al. Reduced release probability prevents vesicle depletion and transmission failure at dynamin mutant synapses. *PNAS*, 2012.

[B17] Takeuchi, H. and Kurahashi, T. Signal transduction in the olfactory receptor cell. In *The Senses: A Comprehensive Reference*. Academic Press, 2008.