HYPER-FOLD: EXPLORING THE EXPRESSIVE LIMIT OF SEQUENCE-GEOMETRY LEARNING FOR PROTEINS VIA HYPERGRAPH MODELING

Anonymous authors
Paper under double-blind review

ABSTRACT

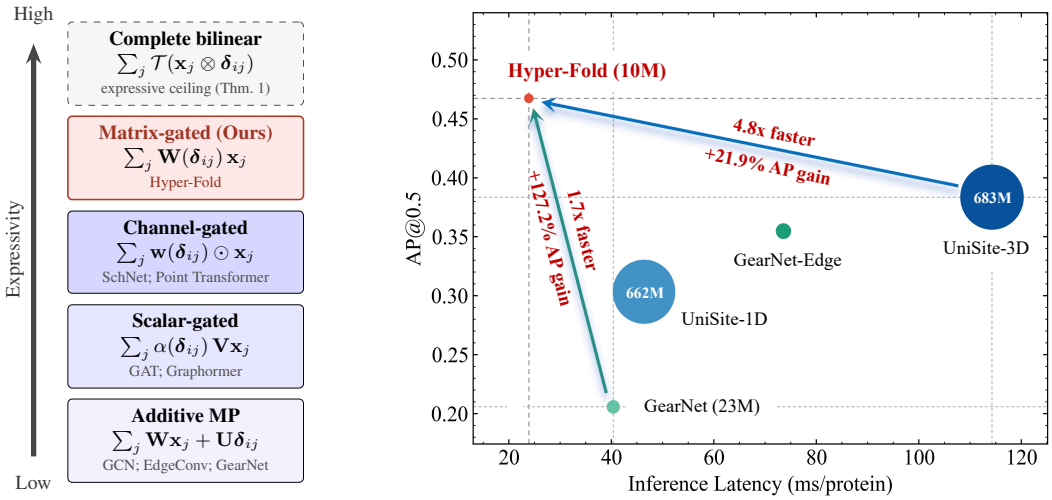
Protein structure modeling rests on a single computational primitive: the interaction between what a residue *is* (sequence content) and *where it sits* (three-dimensional geometry). What is the expressive limit of this layer class? We show that the complete bilinear operator over content–geometry outer products—the sufficient statistic of all second-order interactions—is the expressive ceiling, while the additive message passing of mainstream geometric GNNs is provably blind to content–geometry binding. We then introduce Hyper-Fold, a rank-K separable convolutional backbone approaching this ceiling at message-passing cost: each radius neighborhood is organized into a sequence hyperedge and a contact hyperedge, modulated by an edge-conditioned matrix-valued operator factorized into K learned basis operators with geometry-generated coefficients. Across enzyme function prediction, fold classification, and ligand binding site detection, Hyper-Fold and its hierarchical variant Hyper-Fold-Deep achieve the best results among protein-specific structure encoders; Hyper-Fold-Pocket, an anchored set-prediction head, surpasses UniSite-3D on UniSite-DS and two zero-shot benchmarks with no sequence language model features, $68\times$ fewer parameters, and $4.8\times$ lower latency—suggesting that a sufficiently expressive 3D backbone recovers information that fusion architectures previously borrowed from evolution-scale pretraining.

1 INTRODUCTION

Protein function is determined by three-dimensional structure, and structure modeling is dominated by a single computational primitive: the *interaction* between what a residue *is* (its sequence/content features) and *where it sits* relative to its neighbors (three-dimensional geometry). Message-passing (MP) architectures (Jing et al., 2021; Zhang et al., 2023b; Fan et al., 2023) implement this primitive by propagating geometry-aware messages over residue graphs and remain the most effective structure encoders at modest scale (Jamasb et al., 2024); protein language models and hybrid variants (Lin et al., 2023; Su et al., 2024; Hayes et al., 2025) fold structure into large-scale sequence pretraining. Yet beneath this diversity, no one has systematically answered a basic question: *what is the expressive limit of sequence–geometry interaction layers, and where do existing models sit relative to that limit?* We answer this question: any second-order statistic of a neighbor's content and geometry is a linear functional of their outer product $\mathbf{x}_j \otimes \phi(\boldsymbol{\delta}_{ij})$, so the *complete bilinear operator* over this outer product is the sufficient statistic—and hence the expressive limit—of sequence–geometry interaction (Theorem 1). The result is an expressivity ladder (Figure 1(a)).

Situating existing models on this ladder exposes a structural blind spot. The additive message used by mainstream geometric GNNs, $\mathbf{W}\mathbf{x}_j + \mathbf{U}\boldsymbol{\delta}_{ij}$, factorizes into a content term and a geometry term; consequently it is provably invariant to *binding swaps*—pairs of structures whose node features and edge features are identical as multisets but whose content–geometry pairings differ (Proposition 2). Such binding information is physically real (which residue type occupies which geometric position is precisely what distinguishes a catalytic site from a geometrically identical decoy), yet it is invisible to the lowest rung of the ladder. This gap belongs to the layer class: GearNet explicitly motivated its additive message by the prohibitive cost of the alternative (Zhang et al., 2023b)—cost, not expressivity, was the bottleneck.

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.



(a) The expressivity ladder.

(b) Accuracy vs. Latency on pocket detection task.

Figure 1: Overview. **(a)** The expressivity ladder of sequence–geometry layers: additive message passing occupies the lowest rung, while Hyper-Fold's rank-K matrix gating approaches the complete-bilinear ceiling at message-passing cost. **(b)** Pocket AP@0.5 on UniSite-DS vs. inference latency; bubble area scales with parameter count.

We therefore ask whether the ceiling can be *affordably* approached. Our answer is Hyper-Fold, a rank-K separable convolutional backbone. Inspired by hypergraph modeling (Feng et al., 2019; Yadati et al., 2019), each layer organizes a radius neighborhood into a *sequence hyperedge* and a *contact hyperedge*—sequence adjacency along the chain versus spatial contact in the folded state—and applies to both an edge-conditioned *matrix-valued* operator $\mathbf{W}(\boldsymbol{\delta}_{ij})\,\mathbf{x}_j$, factorized as $\mathbf{W}(\boldsymbol{\delta}) = \sum_{c=1}^{K} g_c(\boldsymbol{\delta})\,\mathbf{B}_c$ with geometry-generated coefficients and learned basis operators. The operator spectrum decays rapidly: $K{=}8$ captures 90–98% of the learned full-operator energy and accuracy saturates beyond it (Proposition 3), so the ceiling is approachable at message-passing cost. The same operator yields two backbones: Hyper-Fold at full residue resolution and Hyper-Fold-Deep for protein-level prediction.

We validate Hyper-Fold on three tasks that span the uses of protein structure encoders: enzyme commission (EC) number prediction, fold classification, and ligand binding site (pocket) detection. Hyper-Fold improves over GearNet-Edge on EC at every sequence-identity cutoff (e.g., 0.864 vs. 0.810 Fmax at the 95% cutoff) and on all three fold-classification splits. For pocket detection we attach Hyper-Fold-Pocket, an anchored set-prediction head on the residue-level backbone: it reaches 0.617 AP@0.3 on UniSite-DS (UniSite-3D: 0.560) and generalizes zero-shot to HOLO4K and COACH420, exceeding UniSite-3D on all six reported metrics, *without any sequence language model features*. This suggests that a sufficiently expressive 3D backbone recovers information that fusion architectures previously borrowed from evolution-scale pretraining.

Our contributions are: (i) an expressivity ladder for sequence–geometry layers, with the complete bilinear operator characterized as its ceiling (Thm. 1) and additive message passing proved blind to content–geometry binding (Prop. 2); (ii) Hyper-Fold, a rank-K separable convolutional backbone that approaches this ceiling at message-passing cost; and (iii) experiments on three tasks, including state-of-the-art pocket detection and zero-shot generalization without sequence-model features, at essentially no extra efficiency cost.

Related work. Pocket detection has evolved from geometry tools (Le Guilloux et al., 2009; Krivák & Hoksza, 2018) to set-prediction formulations (Fan et al., 2025), the formulation Hyper-Fold-Pocket adopts with a purely structural encoder. An extended discussion of structure encoders, sequence models, graph expressivity, and hypergraph modeling is deferred to Appendix A.

2 THE EXPRESSIVE LADDER OF SEQUENCE–GEOMETRY LAYERS

A protein of L residues is a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$: node i carries a *content* feature $\mathbf{x}_i \in \mathbb{R}^C$ (learned residue embeddings), and each ordered edge $(i, j) \in \mathcal{E}$ within a radius cutoff carries a *geometry* feature $\boldsymbol{\delta}_{ij} \in \mathbb{R}^{d_g}$—a 7-dimensional SE(3)-invariant descriptor of the relative placement of residues i and j (distance, direction, and orientation terms)—so all layers below are SE(3)-invariant by construction. A structure encoder stacks layers that update node states $\mathbf{h}_i$ by aggregating over the neighborhood $\mathcal{N}(i)$. We classify such aggregation layers by the *algebraic form* of the interaction between a neighbor's content $\mathbf{x}_j$ and the edge geometry $\boldsymbol{\delta}_{ij}$.

Definition 1 (Five classes of sequence–geometry interaction layers)**.** *With learnable parameters suppressed:*

$$\textit{(additive MP)} \quad \mathbf{h}_i = \textstyle\sum_{j\in\mathcal{N}(i)} \big(\mathbf{W}\mathbf{x}_j + \mathbf{U}\boldsymbol{\delta}_{ij}\big), \tag{1}$$

$$\textit{(scalar-gated)} \quad \mathbf{h}_i = \textstyle\sum_{j\in\mathcal{N}(i)} \alpha(\boldsymbol{\delta}_{ij})\, \mathbf{V}\mathbf{x}_j, \quad \alpha(\cdot) \in \mathbb{R}, \tag{2}$$

$$\textit{(channel-gated)} \quad \mathbf{h}_i = \textstyle\sum_{j\in\mathcal{N}(i)} \mathbf{w}(\boldsymbol{\delta}_{ij}) \odot \mathbf{x}_j, \quad \mathbf{w}(\cdot) \in \mathbb{R}^C, \tag{3}$$

$$\textit{(matrix-gated)} \quad \mathbf{h}_i = \textstyle\sum_{j\in\mathcal{N}(i)} \mathbf{W}(\boldsymbol{\delta}_{ij})\, \mathbf{x}_j, \tag{4}$$

$$\textit{(complete bilinear)} \quad \mathbf{h}_i = \textstyle\sum_{j\in\mathcal{N}(i)} \mathcal{T}\big(\mathbf{x}_j \otimes \phi(\boldsymbol{\delta}_{ij})\big), \tag{5}$$

where α is a scalar gate, $\mathbf{w}$ a per-channel (diagonal) gate, $\mathbf{W}(\cdot)$ a matrix-valued *kernel, ϕ a geometry feature map (the identity, or any richer expansion of $\boldsymbol{\delta}$), and $\mathcal{T}$ an arbitrary linear map on the outer product.*

Class (1) is the message form of GCN, EdgeConv, and GearNet(-Edge) (Kipf & Welling, 2017; Wang et al., 2019; Zhang et al., 2023b); class (2) covers attention-style aggregation (GAT, Graphormer) (Veličković et al., 2018; Ying et al., 2021; Vaswani et al., 2017); class (3) covers continuous-filter convolutions and vector attention (SchNet, Point Transformer) (Schütt et al., 2017; Zhao et al., 2021); classes (4)–(5) are progressively richer.

Proposition 1 (Containment)**.** *Additive MP, scalar-gated, and channel-gated layers are degenerate special cases of matrix-gated layers (after augmenting $\mathbf{x}$ with a constant channel for the bias term): channel gating is $\mathbf{W}(\boldsymbol{\delta})$ restricted to diagonal matrices, scalar gating further ties the diagonal entries into a single factor $\alpha(\boldsymbol{\delta})\mathbf{V}$, and additive MP is recovered by a rank-1, geometry-affine operator. Complete bilinear layers contain matrix-gated layers: expanding the kernel in a basis of the matrix space, $\mathbf{W}(\boldsymbol{\delta})\mathbf{x} = \sum_c g_c(\boldsymbol{\delta})\, \mathbf{B}_c\mathbf{x}$, the map is linear in $\mathbf{x} \otimes \phi(\boldsymbol{\delta})$ whenever the coordinates of ϕ represent the coefficient functions g_c; a universal ϕ represents every continuous kernel (proofs in Appendix F).*

Proposition 2 (Additive layers are blind to content–geometry binding)**.** *Fix a center node i with two neighbors and consider two configurations that differ only by a* binding swap*: configuration A pairs $(\mathbf{x}_1, \boldsymbol{\delta}_1), (\mathbf{x}_2, \boldsymbol{\delta}_2)$ while configuration B pairs $(\mathbf{x}_1, \boldsymbol{\delta}_2), (\mathbf{x}_2, \boldsymbol{\delta}_1)$, with $\mathbf{x}_1 \neq \mathbf{x}_2$ and $\boldsymbol{\delta}_1 \neq \boldsymbol{\delta}_2$. Then every additive MP layer* (1) *produces identical $\mathbf{h}_i$ for A and B, because its output factorizes as $\sum_j \mathbf{W}\mathbf{x}_j + \sum_j \mathbf{U}\boldsymbol{\delta}_{ij}$. In contrast, there exist matrix-gated operators* (4) *that distinguish A from B, e.g. whenever $\mathbf{W}(\boldsymbol{\delta}_1)(\mathbf{x}_1 - \mathbf{x}_2) \neq \mathbf{W}(\boldsymbol{\delta}_2)(\mathbf{x}_1 - \mathbf{x}_2)$ (proof in Appendix F).*

Proposition 2 is not a corner case: additive layers cannot represent *any* function of how content and geometry are paired within a neighborhood—which residue type sits at which geometric position. Gated layers distinguish binding swaps with increasing channel structure—one direction shared by all channels (scalar), C directions that never mix (channel), full cross-channel interaction (matrix)—so the ladder is a hierarchy of how richly content and geometry may be bound.

Theorem 1 (The bilinear ceiling)**.** *Every layer whose message is a second-order (bilinear) function of a neighbor's content $\mathbf{x}_j$ and geometry features $\phi(\boldsymbol{\delta}_{ij})$ factors through the outer product $\mathbf{x}_j \otimes \phi(\boldsymbol{\delta}_{ij})$; conversely, class* (5) *realizes all such functions. Hence the complete bilinear operator is the sufficient statistic of second-order sequence–geometry interaction and constitutes the expressive ceiling of sequence–geometry layers.*

The ceiling is not directly usable: $\mathcal{T}$ has $O(C^2 d_\phi)$ degrees of freedom per edge type. The question becomes how much structure the operator actually has.

Proposition 3 (Rank-K sufficiency, empirical)**.** *Writing the matrix-valued kernel in a learned basis, $\mathbf{W}(\boldsymbol{\delta}) = \sum_{c=1}^{K} g_c(\boldsymbol{\delta})\, \mathbf{B}_c$, we find that the spectrum of the learned full operators decays rapidly:*

measured on the trained K=16 model, the top-8 eigendirections capture 89.9–98.2% of the operator energy in every layer (Appendix F.6), and enlarging K from 8 to 16 leaves pocket AP essentially unchanged. Rank-K matrix gating therefore approaches the ceiling at a small constant cost. A formal approximation bound in terms of the operator spectrum tail is given in Appendix F (Proposition 4).

3 HYPER-FOLD: MATRIX-GATED CONVOLUTION OVER SEQUENCE AND CONTACT HYPEREDGES

Hyper-Fold consists of structure backbones built from a single convolutional operator, Fold-Conv (§3.2, Figure 2), instantiated in two variants (§3.3): Hyper-Fold preserves the sequence length and produces residue-level features, while Hyper-Fold-Deep interleaves residue pooling and produces protein-level features. For pocket detection we attach Hyper-Fold-Pocket, an anchored set-prediction head on the residue-level backbone (§3.4, Figure 3).

3.1 SEQUENCE AND CONTACT HYPEREDGES

A residue relates to its neighborhood through two qualitatively different channels: *sequence adjacency* along the backbone chain and *spatial contact* in the folded state. Inspired by hypergraph modeling (Feng et al., 2019; Yadati et al., 2019)—where a hyperedge connects a center vertex to a *set* of related vertices—we organize the radius neighborhood $\mathcal{N}(i) = \{j : \|\mathbf{p}_i - \mathbf{p}_j\| < r\}$ of each residue i into two hyperedges (Figure 2(a)): a *sequence hyperedge* over the sequence neighbors $\mathcal{N}_{\text{seq}}(i) = \{j \in \mathcal{N}(i) : |s_i - s_j| \le \ell/2\}$ and a *contact hyperedge* over the remaining spatial-contact neighbors $\mathcal{N}_{\text{con}}(i) = \mathcal{N}(i) \setminus \mathcal{N}_{\text{seq}}(i)$, where s_i is the sequence index and ℓ the sequence-window width. We deliberately do *not* adopt hypergraph propagation rules; the hyperedge view only organizes the neighborhood, and both hyperedges are processed by the same matrix-gated convolution below, whose kernel network treats the two hyperedge types with separate embedding weights (§3.2).

The backbone input is a learned residue embedding $\mathbf{E}_{\text{res}} \in \mathbb{R}^{L\times 16}$ of the amino-acid type together with the C_α position matrix $\mathbf{P}_{\text{res}} \in \mathbb{R}^{L\times 3}$; each residue additionally carries a local orientation frame $\mathbf{O}_i \in \mathbb{R}^{3\times 3}$ built from the neighboring C_α trace.

3.2 THE FOLD-CONV LAYER

Geometric feature builder. A layer of radius r connects every ordered pair within $\mathcal{N}(i)$ (plus self-loops) and equips each edge with a 7-dimensional SE(3)-invariant geometry feature and a clamped sequence displacement:

$$\boldsymbol{\delta}_{ij} = \Big[\, \mathbf{O}_j^\top \frac{\mathbf{p}_i - \mathbf{p}_j}{\|\mathbf{p}_i - \mathbf{p}_j\|} \,;\, \textstyle\sum_{k=1}^{3} \mathbf{O}_j^{(k)} \circ \mathbf{O}_i^{(k)} \,;\, \|\mathbf{p}_i - \mathbf{p}_j\| \,\Big] \in \mathbb{R}^7, \;\; \Delta s_{ij} = \text{clamp}(s_i - s_j,\, -\ell/2,\, \ell/2), \tag{6}$$

where $\mathbf{O}^{(k)}$ denotes the k-th frame axis and $\circ$ element-wise multiplication. We abbreviate the three blocks of $\boldsymbol{\delta}_{ij}$ as $(pos_{ij}, ori_{ij}, dis_{ij})$: the relative position in j's local frame, the relative orientation, and the distance (Figure 2(b)). Spatial contact and sequence displacement thus enter the *same* edge object, and the kernel below is conditioned jointly on both—a convolution over fused sequence–contact structure rather than two separate aggregation streams.

Geometric kernel network (GK-Net). The kernel coefficients of each edge are generated by a small geometric kernel network (Figure 2(b)). Its first-layer weights are indexed by a discretized sequence bucket, which is what separates the two hyperedge types: sequence neighbors occupy the ℓ offset buckets, while contact neighbors—whose sequence displacement saturates the clamp—share the boundary bucket:

$$\mathbf{g}(\boldsymbol{\delta}, \Delta s) = \text{MLP}\Big(\text{LReLU}\big(\boldsymbol{\delta}^\top \mathbf{W}_{\text{bucket}(\Delta s)} + \mathbf{b}_{\text{bucket}(\Delta s)}\big)\Big) \in \mathbb{R}^K, \tag{7}$$

with ℓ buckets per layer. Bucketing ties parameters across the geometry space on a coarse sequence grid while $\boldsymbol{\delta}$ modulates within each bucket; a smooth edge gate $\sigma_{ij} \in (0, 1)$ softly downweights edges that are simultaneously far in space and far in sequence (functional form and per-stage ℓ values in Appendix B).

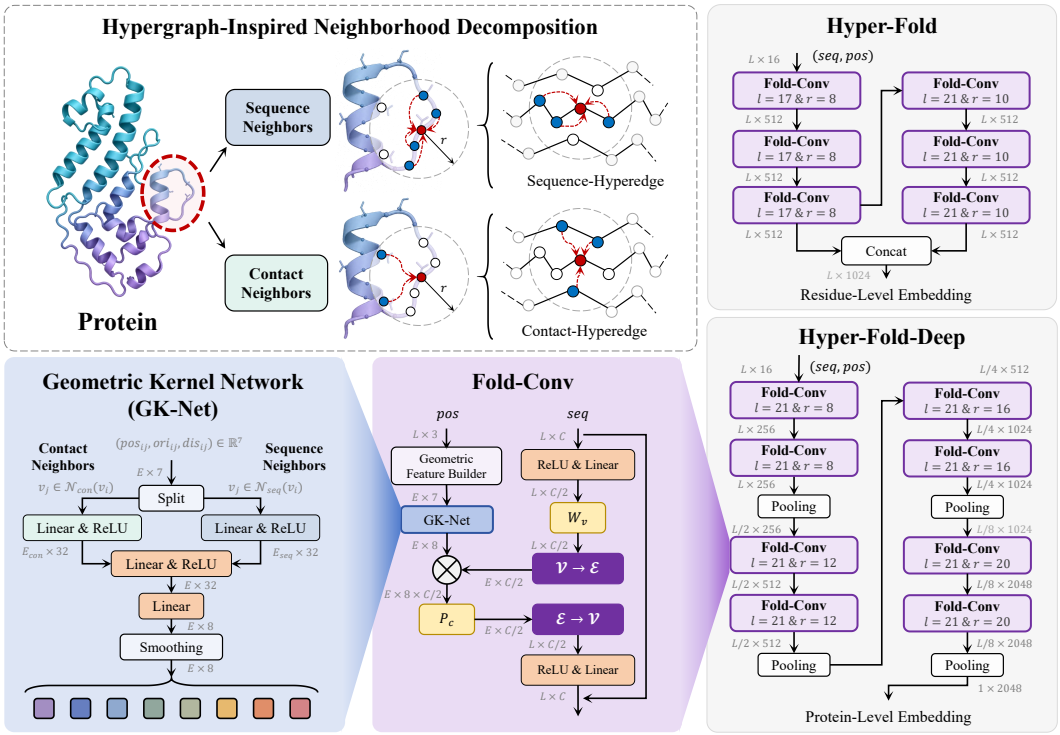


Figure 2: Hyper-Fold backbones. Each residue anchors a *sequence hyperedge* and a *contact hyperedge*; GK-Net maps each edge's geometry feature to rank-K kernel coefficients, modulating neighbor content in a vertex-to-edge / edge-to-vertex convolution. Hyper-Fold produces residue-level embeddings; Hyper-Fold-Deep interleaves residue pooling along a widening channel pyramid. Details in Appendix B.

Outer-product message on hyperedges. Fold-Conv propagates in two steps (Figure 2(c)): a vertex-to-edge step ($\mathcal{V} \to \mathcal{E}$) collects the transformed neighbor content $\mathbf{W}_v \mathbf{x}_j$ onto every edge of both hyperedges, where the rank-K kernel modulates it through an outer product; an edge-to-vertex step ($\mathcal{E} \to \mathcal{V}$) aggregates the readout back to the center residue:

$$\mathbf{h}_i = \sum_{j \in \mathcal{N}(i)} P_c\Big(\sigma_{ij}\, \mathbf{g}(\boldsymbol{\delta}_{ij}, \Delta s_{ij}) \otimes \mathbf{W}_v \mathbf{x}_j\Big), \tag{8}$$

where $\mathbf{g}(\cdot) \in \mathbb{R}^K$ (K=8) are edge-conditioned kernel coefficients, $\sigma_{ij} \in (0,1)$ is the smooth edge gate, $\otimes$ is the outer product, and $P_c : \mathbb{R}^{K \times C} \to \mathbb{R}^{C'}$ is a pointwise readout. Reading P_c channel-wise recovers exactly the matrix-gated form of class (4) (Lemma 1, Appendix F):

$$\mathbf{h}_i = \sum_{j \in \mathcal{N}(i)} \mathbf{W}(\boldsymbol{\delta}_{ij}, \Delta s_{ij})\, \mathbf{x}_j, \qquad \mathbf{W}(\boldsymbol{\delta}, \Delta s) = \sum_{c=1}^{K} \tilde{g}_c(\boldsymbol{\delta}, \Delta s)\, \mathbf{B}_c, \tag{9}$$

with $\tilde{g}_c = \sigma\, g_c$ and learned basis operators $\mathbf{B}_c$ given by the channel blocks of P_c composed with $\mathbf{W}_v$. Eq. (9) is the rank-K factorization anticipated by Proposition 3: a full matrix-valued kernel per edge at $O(K\,C\,C')$ rather than $O(C^2 d_g)$ cost.

The convolution is wrapped in a bottleneck pre-activation residual unit following the GearNet block design (details in Appendix B).

3.3 Two backbone variants

Hyper-Fold (residue-level). Six Fold-Conv blocks are stacked in two stages of three at a fixed width of 512, with a wider sequence window and radius in the deeper stage (Figure 2(d), top; per-stage configurations in Appendix B). A two-scale readout taps the two stage outputs (blocks 3 and 6), each

an $\mathbb{R}^{L\times 512}$ residue-level feature map that preserves the full sequence resolution. Pocket detection consumes both levels separately (§3.4), where every residue must be scored; classification mean-pools each level and concatenates the pooled vectors into a protein embedding in $\mathbb{R}^{1024}$.

Hyper-Fold-Deep (protein-level). For global prediction tasks, Fold-Conv blocks are interleaved with residue pooling along a widening channel pyramid ($256 \to 512 \to 1024 \to 2048$ channels; Appendix B), so the spatial resolution shrinks ($L \to L/2 \to L/4 \to L/8$) while the channel capacity grows (Figure 2(d), bottom); a final global pooling produces a protein-level embedding $\mathbf{f}_{\text{prot}} \in \mathbb{R}^{2048}$ that feeds an MLP classifier for EC and fold classification.

3.4 Hyper-Fold-Pocket: anchored set prediction for pockets

Pocket detection is cast as set prediction (Carion et al., 2020): the model outputs a *set* of pockets, each a confidence score paired with a per-residue mask, without post-processing clustering. Hyper-Fold-Pocket builds on the residue-level Hyper-Fold backbone—every residue must be scored, so the pooling pyramid of Hyper-Fold-Deep does not apply—and follows a four-stage pipeline (Figure 3): a neck projects the two backbone levels into query, key, and memory features; a pocket proposal network (PPN) selects structure-anchored queries; a transformer decoder refines them over the memory; and two heads read out each pocket. Relative to the DETR paradigm, the design is decoder-only, anchors its queries on the actual 3-D structure, and predicts masks instead of boxes (detailed comparisons with DETR-style detectors and UniSite are in Appendix B).

Structure-anchored proposals over a bi-level memory. Instead of learning the queries as free parameters, a linear *pocketness* probe over the fused residue features scores every residue, and the top-N_q (N_q=50) become proposals: their own residue features initialize the query content, and their C_α coordinates, embedded by a three-dimensional Fourier positional encoding (3D-PE), provide the query position—in the spirit of anchored query selection in RT-DETR (Lv et al., 2024), but with anchors that are actual residues. The decoder memory is likewise structure-preserving: the two backbone levels, tagged by learnable level embeddings, are stacked into a bi-level memory $\mathbf{M}_{\text{emb}} \in \mathbb{R}^{2L\times 256}$, which a 4-layer transformer decoder cross-attends to directly—with no interposed transformer encoder—injecting the 3D-PE at every layer.

Prediction and training. Each refined query $\mathbf{q}_k$ is read out by a classification head (pocket / no-pocket) and a mask head that takes the inner product with the neck's residue keys, $\mathbf{m}_k = \sigma\big(\langle \text{MLP}_{\text{mask}}(\mathbf{q}_k),\, \mathbf{K}_{\text{emb}} \rangle\big) \in [0,1]^L$, so every proposal independently predicts its own residue mask. Training matches proposals to ground-truth pockets by Hungarian matching and sums classification,

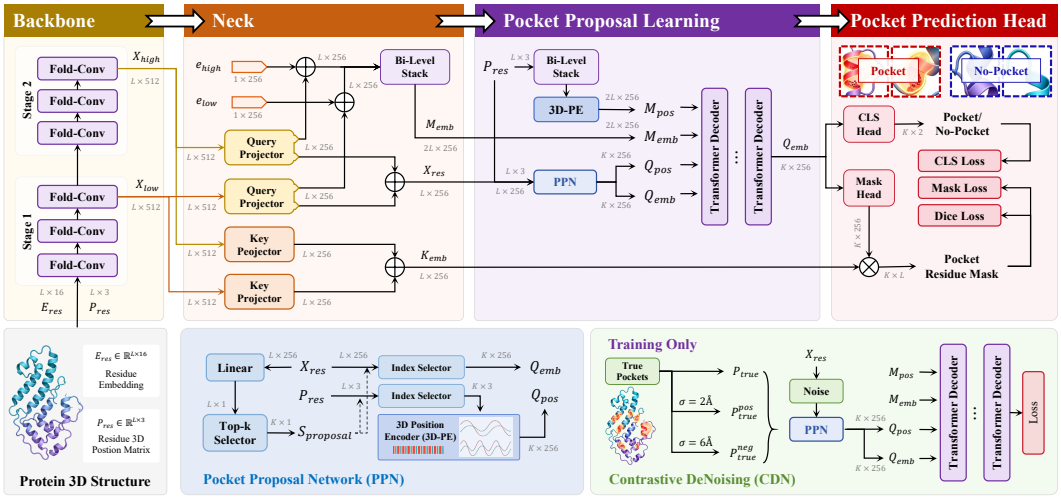


Figure 3: Hyper-Fold-Pocket pipeline. Two backbone feature levels feed a query/key/memory neck; the PPN selects structure-anchored queries, refined by a 4-layer transformer decoder over the bi-level memory; a classification head and a mask head then read out each pocket. Details in Appendix B.

per-residue mask, and Dice losses, with auxiliary supervision at every decoder layer (matching weights in Appendix B). Contrastive denoising (CDN) training (Li et al., 2022) appends denoising queries built from ground-truth pockets—two positive copies (one clean, one lightly noised) and one heavily noised *contrastive negative* labeled as no-object, attention-isolated from the real queries—which stabilizes bipartite matching in early training and teaches the decoder to reject near-miss decoys (noise scales in Appendix B). The same head and training configuration is shared by all backbone comparisons of §4.

4 EXPERIMENTS

4.1 SETUP

Tasks and data. EC number prediction and fold classification follow the GearNet protocol and splits (Zhang et al., 2023b); we report Fmax@50% and AUPR@95 for EC in the main text (all five cutoffs in Appendix D.1) and accuracy on the three fold-classification splits (fold / superfamily / family). Pocket detection is trained on UniSite-DS (Fan et al., 2025) and evaluated in-distribution (AP at IoU 0.3/0.5) and zero-shot on HOLO4K-sc and COACH420 (AP@0.3, DCC, DCA), matching the UniSite evaluation protocol.

Baselines and training. Classification baselines are GCN, GAT, Point Transformer, SchNet, GVP, GearNet(-Edge/-IEConv), ProNet(-Backbone), CDConv, and SCHull; values follow Zhang et al. (2023b, Tables 2, 6 and 7) where available and are our reproductions under the unified recipe otherwise (provenance and deviations in Appendix C). Pocket baselines are Fpocket, Fpocket-rescore, P2Rank, DeepPocket, GrASP, VN-EGNN, and UniSite-1D/3D (Fan et al., 2025). All models use AdamW with warmup and cosine decay, EMA, mixed precision, and gradient clipping; full recipes, hardware, and the latency protocol are in Appendix C.

4.2 MAIN RESULTS

Table 1: EC number prediction and fold classification results. EC reports Fmax at the 50% sequence-identity cutoff and AUPR at 95% (full table in Appendix D.1); fold reports test accuracy on three held-out splits. Rows above/below the rule are generic/protein-specific architectures. **Bold**: best; underlined: second best (ties included).

			EC Prediction		Fold Classification		
Method	Params	Latency	Fmax@50%	AUPR	Fold	Superfamily	Family
GCN	21.9M	6.2 ms	0.246	0.319	0.168	0.213	0.828
GAT	21.9M	6.9 ms	0.247	0.339	0.124	0.165	0.727
Point Transformer	14.5M	9.7 ms	0.276	0.346	0.181	0.211	0.689
SchNet	6.8M	8.1 ms	0.617	0.714	0.221	0.319	0.863
GVP	7.1M	10.0 ms	0.347	0.467	0.274	0.459	0.938
GearNet	31.1M	4.0 ms	0.615	0.751	0.284	0.426	0.953
GearNet-Edge	40.7M	48.6 ms	0.694	0.835	0.440	0.667	0.991
GearNet-Edge-IEConv	46.4M	44.1 ms	0.718	0.831	0.483	0.703	<u>0.995</u>
ProNet	1.5M	9.1 ms	0.669	0.795	0.462	0.553	0.947
ProNet-Backbone	4.3M	11.9 ms	0.629	0.755	0.527	0.703	0.993
CDConv	31.0M	14.2 ms	0.777	0.859	<u>0.568</u>	0.765	<u>0.995</u>
SCHull	4.3M	11.9 ms	0.664	0.787	0.561	0.746	0.994
Hyper-Fold (ours)	7.7M	10.5 ms	**0.789**	**0.874**	0.540	<u>0.775</u>	**0.996**
Hyper-Fold-Deep (ours)	20.6M	15.7 ms	<u>0.787</u>	<u>0.873</u>	**0.578**	**0.794**	<u>0.995</u>

Function and structure classification. Table 1 compares Hyper-Fold with from-scratch structure encoders. Hyper-Fold-Deep attains the best accuracy among protein-specific methods on the Fold split—the hardest generalization setting—surpassing even the IEConv-augmented GearNet variant with a single convolution type instead of a heterogeneous layer stack, and is also best on the Superfamily split (Family is near-saturated for all recent methods). On EC, Hyper-Fold attains the best numbers, with Hyper-Fold-Deep close behind—both ahead of every baseline. The two variants occupy complementary sweet spots: the flat Hyper-Fold preserves the residue-level detail that

Table 2: Pocket detection: in-distribution (UniSite-DS) and zero-shot generalization (HOLO4K-sc, COACH420; all methods trained on UniSite-DS only). Left: AP at IoU 0.3/0.5; right: zero-shot AP@0.3 and DCC/DCA top-n success rates (4 Å). **Bold**: best; underlined: second best.

	UniSite-DS		HOLO4K-sc (zero-shot)			COACH420 (zero-shot)		
Method	AP@0.3 ↑	AP@0.5 ↑	AP@0.3 ↑	DCC ↑	DCA ↑	AP@0.3 ↑	DCC ↑	DCA ↑
Fpocket	0.184	0.102	0.271	0.308	0.438	0.211	0.271	0.411
Fpocket-rescore	0.508	0.235	0.590	0.518	0.765	0.560	0.441	0.711
P2Rank	0.506	0.216	0.601	0.530	0.819	0.619	0.464	0.741
DeepPocket	0.427	0.233	0.542	0.493	0.737	0.518	0.396	0.676
GrASP	0.447	0.285	0.667	0.513	0.742	0.715	0.485	0.762
VN-EGNN	0.162	0.071	0.261	0.586	0.700	0.264	0.545	0.753
UniSite-1D	0.512	0.303	0.687	0.554	0.769	0.592	0.455	0.735
UniSite-3D	0.560	0.384	0.709	0.572	0.788	0.720	0.470	0.738
Hyper-Fold-Pocket (ours)	**0.617**	**0.467**	**0.735**	**0.681**	**0.827**	**0.768**	**0.563**	**0.786**

function annotation rewards, while the hierarchical Hyper-Fold-Deep turns multi-scale pooling into the best global fold representation—at 19–21M parameters, both remain smaller than the GearNet family and CDConv.

Ligand binding site detection. We have three observations. First, Hyper-Fold-Pocket sets the best in-distribution result on UniSite-DS (Table 2), improving AP@0.3 by 5.7 points over UniSite-3D and AP@0.5 by 8.3 points. Second, the gap widens out of distribution: Hyper-Fold-Pocket exceeds UniSite-3D on all six zero-shot metrics, with the largest gain on localization quality (DCC: +10.9 on HOLO4K-sc, +9.3 on COACH420). Third, all of this is achieved with *pure three-dimensional input*: no ESM features, no evolutionary information.

4.3 The expressivity ladder is causal: operator-only swaps

Table 3: Operator-only ablation: all rows share the default Hyper-Fold backbone, task head, and training recipe; only the aggregation operator varies. Params and latency refer to the classification configuration. **Bold**: best; underlined: second best.

			EC prediction		Pocket detection	
Operator	Params	Latency	Fmax@50%	AUPR	AP@0.3	AP@0.5
Additive MP	3.9M	9.6 ms	0.677	0.808	0.534	0.264
Scalar gate	3.9M	11.2 ms	0.713	0.834	0.500	0.247
Channel gate	3.6M	10.3 ms	0.777	0.872	0.608	0.432
Matrix gate (Hyper-Fold)	6.6M	10.5 ms	**0.789**	**0.874**	**0.617**	**0.467**

Table 3 makes the expressivity ladder empirical. The two lowest rungs sit close together, in task-dependent order—a single geometry-conditioned coefficient per edge, however computed, cannot bind content to geometry. The decisive jump comes from *per-channel* modulation, which closes most of the gap on EC; yet the full matrix gate still leads on every metric of both tasks, its margin over channel gating growing at the stricter IoU threshold (3.5 AP@0.5 points)—precisely where content–geometry binding should matter most.

4.4 Analysis

Figure 1(b) plots measured latency against accuracy, marker size proportional to parameters: Hyper-Fold-Pocket (10.07M, 23.9 ms) is $4.8\times$ faster and $68\times$ smaller than UniSite-3D (683.9M, 114.3 ms) at the best accuracy, and the ESM-2 embedding stage alone (38.1 ms) costs more than the entire Hyper-Fold-Pocket forward pass—consistent with the theory, as Fold-Conv keeps the $O(|\mathcal{E}|\,C^2)$ order of additive message passing whereas transformer-based pipelines pay $O(L^2)$ self-attention over residues (complexity analysis in Appendix B). The gap is not only asymptotic: the complete bilinear operator is memory-infeasible at residue resolution, and rank-K factorization is what makes

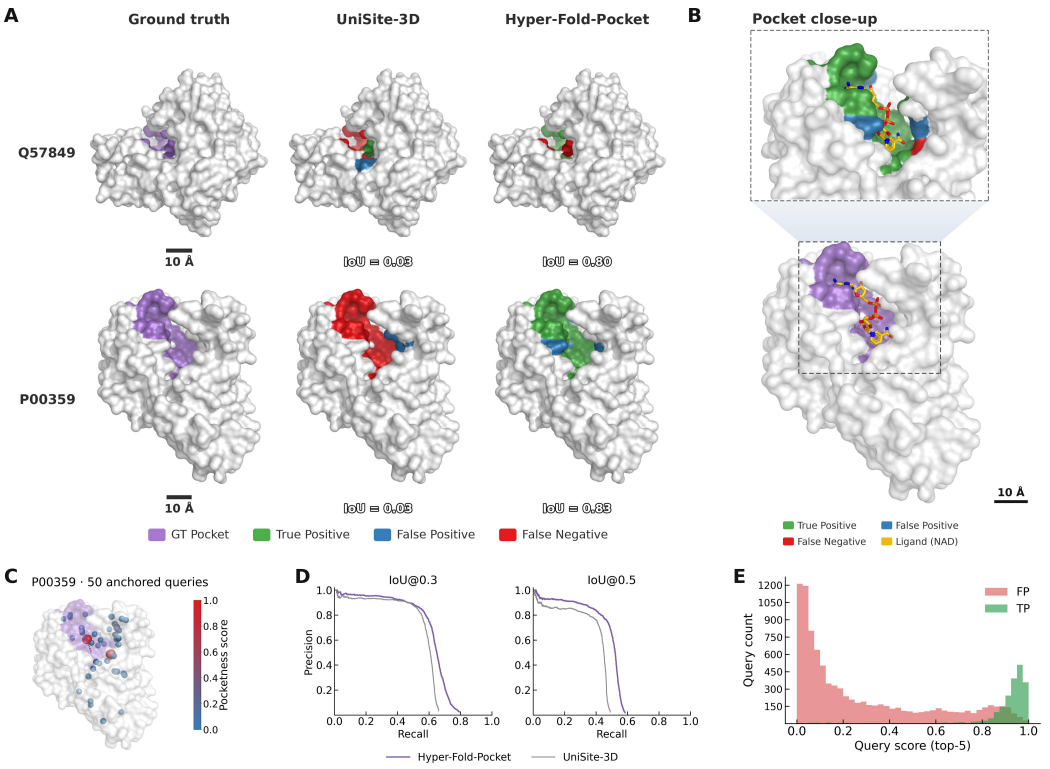


Figure 4: Qualitative analysis of ligand-binding-site detection. **(A)** Per-residue error maps on Q57849 and P00359 (rows; columns: ground truth, UniSite-3D, Hyper-Fold-Pocket): pocket in purple; TP/FP/FN in green/blue/red; pocket IoU under each prediction. **(B)** Pocket anatomy on P00359: whole protein with ground-truth pocket in purple and the bound NAD ligand; the close-up uses the same error coloring. **(C)** Pocketness distribution of the 50 anchored queries. **(D)** Precision–recall curves at IoU 0.3/0.5. **(E)** Top-5 query scores: high scores are almost exclusively true positives. Panel protocols in Appendix E.

the ceiling usable there. In pocket ablations (Appendix E), the rank-K kernel saturates at $K{=}8$ and contrastive denoising adds 1.5 AP@0.5.

Figure 4 dissects where the detection gain comes from: UniSite-3D's errors scatter into false-positive surface patches and missed pocket residues, whereas Hyper-Fold-Pocket's align tightly with the ground truth (Figure 4A–B); the anchored queries concentrate their pocketness on pocket neighborhoods (Figure 4C); and the top-5 query scores are almost exclusively true positives (Figure 4E), mirroring the PR-curve dominance at both IoU thresholds (Figure 4D). On multi-pocket proteins, anchored queries specialize to distinct sites rather than competing for one. The margin persists under distribution shift: center-localization gains are largest on the zero-shot benchmarks (DCC +9–11 over UniSite-3D), suggesting expressive 3D features transfer better than language-model fusion.

5 Conclusion

The expressive limit of sequence–geometry interaction has a clean answer: the complete bilinear operator over content–geometry outer products. Mainstream additive message passing occupies the lowest rung and is provably blind to content–geometry binding; Hyper-Fold approaches the ceiling at message-passing cost, with payoff spanning function annotation, fold classification, and zero-shot pocket detection—without sequence language model features.

AI USE STATEMENT

In this work, we used generative AI tools for literature search and retrieval during background research, for assistance in checking and testing code, and for language polishing of the manuscript. All AI-assisted outputs were reviewed and verified by the authors, and all quantitative results were read directly from raw experimental logs. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies protein structure modeling on public benchmarks; it involves no human subjects, sensitive or personally identifiable data, and no foreseeable harmful applications. The authors have read and adhere to the ICLR Code of Ethics.

REPRODUCIBILITY STATEMENT

Appendix B specifies all architecture configurations and Appendix C the full training recipes; complete proofs of all theoretical claims are given in Appendix F. Anonymized source code, configuration files, and trained model weights are included in the supplementary material; the pocket model runs on a single PDB file out of the box.

REFERENCES

Rishal Aggarwal, Akash Gupta, Vineeth Chelur, C V Jawahar, and U Deva Priyakumar. DeepPocket: ligand binding site detection and segmentation using 3d convolutional neural networks. *Journal of Chemical Information and Modeling*, 62(21):5069–5079, 2021.

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

Hehe Fan, Zhangyang Gao, Yi Kuo, Lin Li, and Stan Z Li. Continuous-discrete convolution for geometry-guided modeling of proteins and point clouds. In *International Conference on Learning Representations*, 2023.

Jigang Fan, Quanlin Wu, Shengjie Luo, and Liwei Wang. UniSite: The first cross-structure dataset and learning framework for end-to-end ligand binding site detection. *arXiv preprint arXiv:2506.03237*, 2025.

Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. Hypergraph neural networks. In *AAAI Conference on Artificial Intelligence*, 2019.

Yifan Feng, Jiashu Han, Shihui Ying, and Yue Gao. Hypergraph isomorphism computation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(5):3880–3896, 2024.

Yifan Feng, Jiangang Huang, Shaoyi Du, Shihui Ying, Jun-Hai Yong, Yipeng Li, Guiguang Ding, Rongrong Ji, and Yue Gao. Hyper-YOLO: When visual object detection meets hypergraph computation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47, 2025.

Yifan Feng, Rizhuo Huang, Yifan Zhang, Shaoyi Du, Shihui Ying, Zongze Wu, and Yue Gao. How powerful are hypergraph neural networks? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2026.

Yue Gao, Yifan Feng, Shuyi Ji, and Rongrong Ji. HGNN$^+$: General hypergraph neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):3181–3199, 2023.

Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. DeepFM: a factorization-machine based neural network for CTR prediction. In *International Joint Conference on Artificial Intelligence*, 2017.

Tomas Hayes, Roshan Rao, Halil Akin, Nicholas Sofroniew, et al. Simulating 500 million years of evolution with a language model. *Science*, 2025.

Arian R Jamasb, Alex Morehead, Rishabh Malhotra, Kieran Didi, Pietro Sormanni, et al. Evaluating representation learning on the protein structure universe. In *International Conference on Learning Representations*, 2024.

Bowen Jing, Stephan Eismann, Patricia Suriana, Raphael John Lamarre Townshend, and Ron Dror. Learning from protein structure with geometric vector perceptrons. In *International Conference on Learning Representations*, 2021.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

Radoslav Krivák and David Hoksza. P2Rank: machine learning based tool for rapid and accurate prediction of ligand binding sites from protein structure. *Journal of Cheminformatics*, 10:1–12, 2018.

Vincent Le Guilloux, Peter Schmidtke, and Pierre Tuffery. Fpocket: An open source platform for ligand pocket detection. *BMC Bioinformatics*, 10:168, 2009.

Feng Li, Hao Zhang, Shilong Liu, Jian Guo, Lionel M Ni, and Lei Zhang. DN-DETR: Accelerate DETR training by introducing query denoising. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.

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

Wenyu Lv, Shangliang Xu, Yian Zhao, Guanzhong Wang, Jiaman Wei, Cheng Cui, Yuning Du, Qingqing Dang, and Yi Liu. DETRs beat YOLOs on real-time object detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Haggai Maron, Heli Ben-Hamu, Nadav Shamir, and Yaron Lipman. Invariant and equivariant graph networks. In *International Conference on Machine Learning*, 2019.

Alex Morehead and Jianlin Cheng. Geometry-complete perceptron networks for 3d molecular graphs. *Bioinformatics*, 2024.

Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C Lawrence Zitnick, Jerry Ma, et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, 118(15), 2021.

Kristof T Schütt, Pieter-Jan Kindermans, Huziel E Sauceda, Stefan Chmiela, Alexandre Tkatchenko, and Klaus-Robert Müller. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. In *Advances in Neural Information Processing Systems*, 2017.

Florian Sestak, Lisa Schneckenreiter, Johannes Brandstetter, Sepp Hochreiter, Andreas Mayr, and Günter Klambauer. VN-EGNN: E(3)-equivariant graph neural networks with virtual nodes enhance protein binding site identification. *arXiv preprint arXiv:2404.07194*, 2024.

Zachary Smith, Michael Strobel, Bodhi P Vani, and Pratyush Tiwary. Graph attention site prediction (GrASP): identifying druggable binding sites using graph neural networks with attention. *Journal of Chemical Information and Modeling*, 64(7):2637–2644, 2024.

Jin Su, Chenchen Han, Yuyang Zhou, Junjie Shan, Xibin Zhou, and Fajie Yuan. SaProt: Protein language modeling with structure-aware vocabulary. In *International Conference on Learning Representations*, 2024.

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*, 2017.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

Limei Wang, Haoran Liu, Yi Liu, Jerry Kurtin, and Shuiwang Ji. Learning hierarchical protein representations via complete 3D graph networks. In *International Conference on Learning Representations*, 2023.

Shih-Hsin Wang, Yuhao Huang, Justin Baker, et al. A theoretically-principled sparse, connected, and rigid graph representation of molecules. In *International Conference on Learning Representations*, 2025.

Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics*, 38(5):1–12, 2019.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019.

Naganand Yadati, Madhav Nimishakavi, Prateek Yadav, Vikram Nitin, Anand Louis, and Partha Talukdar. HyperGCN: A new method of training graph convolutional networks on hypergraphs. In *Advances in Neural Information Processing Systems*, 2019.

Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? In *Advances in Neural Information Processing Systems*, 2021.

Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel Ni, and Heung-Yeung Shum. DINO: DETR with improved denoising anchor boxes for end-to-end object detection. In *International Conference on Learning Representations*, 2023a.

Zuobai Zhang, Minghao Xu, Arian Jamasb, Vijil Chenthamarakshan, Aurélie Lozano, Payel Das, and Jian Tang. Protein representation learning by geometric structure pretraining. In *International Conference on Learning Representations*, 2023b.

Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *IEEE/CVF International Conference on Computer Vision*, 2021.

Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: Deformable transformers for end-to-end object detection. In *International Conference on Learning Representations*, 2021.

A Extended related work

Geometric deep learning on protein structures. Structure-based encoders represent a protein as a residue graph and learn with SE(3)-invariant or equivariant layers: GVP (Jing et al., 2021) introduced geometric vector perceptrons; GearNet (Zhang et al., 2023b) added edge message passing and remains a strong multi-task baseline (Jamasb et al., 2024); CDConv (Fan et al., 2023) and GCPNet (Morehead & Cheng, 2024) explore continuous or geometry-complete alternatives. §2 stratifies these layers by their gating algebra; here we only note the closest point in expressivity, CDConv, whose matrix-valued kernels match ours in principle but are memory-intensive at full residue resolution, forcing early downsampling that trades away exactly the residue-level detail tasks such as pocket detection require. Hyper-Fold keeps the matrix-gated kernel but factorizes it at rank K over sequence and contact hyperedges, bringing the cost into the message-passing regime at full sequence length—$4\times$ fewer parameters and lower latency than CDConv (Table 1)—which is precisely what makes the residue-level Hyper-Fold-Pocket head practical.

Sequence models and sequence–structure fusion. Protein language models pretrained at evolution scale (Rives et al., 2021; Lin et al., 2023) dominate sequence-based prediction, and hybrid models inject structure into them (Su et al., 2024; Hayes et al., 2025). Our results suggest that a sufficiently expressive 3D backbone can make such fusion unnecessary (§4.2).

Ligand binding site detection. Geometry methods (Fpocket (Le Guilloux et al., 2009)), ML rescoring (P2Rank (Krivák & Hoksza, 2018)), and CNN/GNN detectors (Aggarwal et al., 2021; Smith et al., 2024; Sestak et al., 2024) operate on surface or graph features; UniSite (Fan et al., 2025) reframed detection as set prediction over fused sequence–structure features. We adopt the set-prediction formulation but replace the encoder.

Expressivity of graph networks. The WL hierarchy bounds standard MPNNs (Xu et al., 2019), and invariant tensor networks characterize maximal universal layers (Maron et al., 2019); on hypergraphs, the expressive power of hypergraph neural networks has been characterized via generalized Weisfeiler–Lehman tests (Feng et al., 2026). Our analysis is orthogonal and finer-grained: within the *geometry-conditioned* layers actually used on proteins, we stratify expressivity by the algebraic form of the content–geometry interaction, in the spirit of factorization-machine analyses of feature interaction (Guo et al., 2017).

Hypergraph modeling. Hypergraph neural networks (Feng et al., 2019; Yadati et al., 2019) generalize pairwise edges to hyperedges that connect a vertex to a *set* of related vertices, with HGNN$^+$ (Gao et al., 2023) unifying vertex- and hyperedge-level propagation, and hypergraph isomorphism computation (Feng et al., 2024) characterizing structural equivalence of such high-order structures. Hypergraph modeling has also proven effective in detection: Hyper-YOLO (Feng et al., 2025) constructs hypergraphs over visual features to capture high-order correlations among candidate regions. Hyperedges are a natural language for the two residue groupings we use—sequence adjacency and spatial contact. Hyper-Fold borrows this grouping view only: each residue anchors a sequence hyperedge and a contact hyperedge, but information propagation is carried out by our own matrix-gated convolution (§3.2), not by hypergraph propagation rules.

B Architecture details

B.1 Hyper-Fold backbone

The backbone maps the residue-level input (amino-acid embedding, C_α coordinates, and local frames; §3.1) to residue- or protein-level features by stacking Fold-Conv blocks on the radius graph with its two hyperedge types. Every block runs the same pipeline: the geometric feature builder computes $\boldsymbol{\delta}_{ij}$ and Δs_{ij} for each edge (Eq. (6)), the geometric kernel network (GK-Net) turns them into rank-K kernel coefficients (Eq. (7)), the outer-product message of Eq. (8) aggregates over both hyperedges, and a bottleneck residual unit closes the block. The two variants differ only in how the blocks are staged (§3.3). The remaining implementation details follow.

Fold-Conv residual block. The convolution of Eq. (8) is wrapped in a bottleneck pre-activation residual unit, $\mathbf{h}^{(b+1)} = \mathbf{h}^{(b)} + W_{\text{out}}\,\mathrm{Drop}\big(\mathrm{Conv}_{r_b}\big(\mathrm{LReLU}(\mathrm{BN}(W_{\text{in}}\mathbf{h}^{(b)}))\big)\big)$, with bottleneck width $C/2$ and BN momentum 0.2, following the GearNet block design.

GK-Net bucketing and smooth gate. The bucketed first layer of GK-Net (Eq. (7)) uses ℓ offset buckets per stage (ℓ=17 in the shallow stage of Hyper-Fold, ℓ=21 in the deeper stage and in Hyper-Fold-Deep, which uses ℓ=5 for fold); contact neighbors, whose sequence displacement saturates the clamp, share the boundary bucket. The smooth edge gate is $\sigma_{ij} = \frac{1}{2}\big[1 - \tanh\big(16\,\bar{d}\,\bar{s} - 14\big)\big]$ with $\bar{d} = \|\mathbf{p}_i - \mathbf{p}_j\|/r$ and $\bar{s} = |\Delta s_{ij}|/(\ell/2)$, softly downweighting edges that are simultaneously far in space and far in sequence.

Backbone configurations. Hyper-Fold stacks six Fold-Conv blocks in two stages of three blocks each, with $(\ell, r) = (17, 8\,\text{Å})$ and $(21, 10\,\text{Å})$ respectively, at a constant width of 512 (bottleneck blocks with base width 32). The readout taps the two stage outputs (blocks 3 and 6): classification mean-pools each level and concatenates the pooled vectors into $\mathbb{R}^{1024}$, while the pocket neck consumes the two levels separately (the Hyper-Fold-Pocket paragraph below). Hyper-Fold-Deep is a hierarchical variant of Hyper-Fold with a four-stage widening pyramid of channels $256 \to 512 \to 1024 \to 2048$: each stage contains two Fold-Conv blocks with radius $r = 8, 12, 16, 20$ Å per stage, followed by residue pooling that progressively shortens the sequence (19–21M parameters depending on the task head). The sequence window follows the per-task convention of the reference protocol: $\ell = 21$ for

EC and $\ell = 5$ for fold, where we find a wide window overfits (test fold accuracy $0.578 \to 0.493$ at ℓ=21).

Complexity. Per Fold-Conv layer, evaluating the rank-K kernel of Eq. (9) costs $O(|\mathcal{E}|\, K\, C\, C')$ with bottleneck width C'=$C/2$ and small constant K=8—the same order as additive message passing's $O(|\mathcal{E}|\, C^2)$, up to a small constant factor. The graph itself is sparse: each residue aggregates from a fixed window of ℓ sequence neighbors plus its radius neighbors, whose count is bounded by the near-constant packing density of C_α atoms, so $|\mathcal{E}| = O\big(L(\ell{+}\rho)\big)$ with ℓ, ρ independent of L and the per-layer cost grows *linearly* with sequence length. Transformer-based pipelines such as UniSite instead pay $O(L^2 d)$ self-attention over all residue pairs on top of the backbone. The pocket head preserves the linear scaling: with the proposal count N_q fixed, decoder cross-attention over the $2L$ memory rows and the mask inner products both cost $O(L\, N_q\, d)$. Measured latency confirms the gap to additive message passing is a small constant factor, while remaining orders of magnitude cheaper than protein-language-model-based pipelines (§4.4).

B.2 HYPER-FOLD-POCKET

Hyper-Fold-Pocket turns the residue-level features of Hyper-Fold into a set of pocket predictions in four stages (§3.4): a neck projects the two backbone levels into query, key, and memory features; the pocket proposal network (PPN) scores every residue and selects the top-N_q structure-anchored proposals; a 4-layer transformer decoder refines them against the bi-level memory; and two heads produce, per proposal, a pocket score and a residue mask. Training combines Hungarian-matched classification/mask/Dice losses with contrastive denoising. The complete design follows.

Neck. The residue-level backbone runs at width 512 and is read out at blocks $\{3, 6\}$, giving $\mathbf{X}_{\text{low}}, \mathbf{X}_{\text{high}} \in \mathbb{R}^{L\times 512}$; its node input is a learned 16-dimensional amino-acid-type embedding. Because the structure may not cover the full UniProt sequence, backbone features are scattered onto the UniProt residue axis of length L: positions without structural coverage receive a shared learned null vector, and a binary mask records coverage. Two pairs of linear projectors then produce the fused residue features and the mask keys,

$$\mathbf{X}_{\text{res}} = \mathbf{X}_{\text{low}}\mathbf{W}^q_{\text{low}} + \mathbf{X}_{\text{high}}\mathbf{W}^q_{\text{high}}, \qquad \mathbf{K}_{\text{emb}} = \mathbf{X}_{\text{low}}\mathbf{W}^k_{\text{low}} + \mathbf{X}_{\text{high}}\mathbf{W}^k_{\text{high}}, \tag{10}$$

with $\mathbf{W}^q_{\cdot}, \mathbf{W}^k_{\cdot} \in \mathbb{R}^{512\times 256}$, both outputs in $\mathbb{R}^{L\times 256}$. For the decoder memory, each level's query projection is tagged with a learnable level embedding $\mathbf{e}_{\text{low}}, \mathbf{e}_{\text{high}} \in \mathbb{R}^{256}$ and the two levels are concatenated along the residue axis into the bi-level memory $\mathbf{M}_{\text{emb}} = [\,\mathbf{X}_{\text{low}}\mathbf{W}^q_{\text{low}} + \mathbf{e}_{\text{low}};\ \mathbf{X}_{\text{high}}\mathbf{W}^q_{\text{high}} + \mathbf{e}_{\text{high}}\,] \in \mathbb{R}^{2L\times 256}$, paired with a shared three-dimensional Fourier positional encoding (3D-PE) $\mathbf{M}_{\text{pos}}$ (the same residue coordinates, repeated for both levels) and a padding mask that excludes invalid and unmapped positions.

PPN. A linear probe $s_i = \mathbf{w}^\top \mathbf{x}_{\text{res},i}$ with $\mathbf{w} \in \mathbb{R}^{256}$ scores every residue (*pocketness*), supervised by a residue-level binary cross-entropy against the union of ground-truth masks (loss weight 2). The top-N_q eligible residues (N_q=50; eligible = structurally mapped and valid) become proposals: their rows of $\mathbf{X}_{\text{res}}$ initialize the query content $\mathbf{Q}_{\text{emb}} \in \mathbb{R}^{N_q\times 256}$, and their C_α coordinates serve as anchors $\{\mathbf{a}_k\}_{k=1}^{N_q}$, embedded by the 3D-PE below into $\mathbf{Q}_{\text{pos}}$. Proteins with fewer than N_q eligible residues pad the remaining queries with a learned null content vector.

3D Fourier positional encoding. All coordinates are protein-centered (each structure's C_α centroid is subtracted), so the encoding is translation-invariant like the backbone. A coordinate $\mathbf{a} \in \mathbb{R}^3$ is embedded per axis with 8 frequency bands, $\gamma(\mathbf{a}) = [\sin(2^m\mathbf{a}), \cos(2^m\mathbf{a})]_{m=0}^{7} \in \mathbb{R}^{48}$, followed by a learned linear projection to 256 dimensions. This yields the query positional encoding $\mathbf{Q}_{\text{pos}}$ (of the anchors) and the memory positional encoding $\mathbf{M}_{\text{pos}}$ (of every residue).

Decoder. A 4-layer transformer decoder (8 attention heads, feed-forward width 1024, dropout 0.1, post-norm with a final LayerNorm) refines $\mathbf{Q}_{\text{emb}}$ by cross-attending to the bi-level memory $(\mathbf{M}_{\text{emb}}, \mathbf{M}_{\text{pos}})$, with $\mathbf{Q}_{\text{pos}}$ injected as the query positional encoding at every layer. The output of each decoder layer is read out by the prediction heads below, providing auxiliary supervision.

Prediction heads. Each refined query $\mathbf{q}_k$ is read out by a two-layer classification MLP $z_k = \mathrm{MLP}_{\mathrm{cls}}(\mathbf{q}_k) \in \mathbb{R}^2$ (pocket / no-pocket) and a three-layer mask MLP followed by an inner product with the mask keys, $\mathbf{m}_k = \sigma\big(\langle \mathrm{MLP}_{\mathrm{mask}}(\mathbf{q}_k),\, \mathbf{K}_{\mathrm{emb}} \rangle\big) \in [0,1]^L$, so every proposal independently predicts its own residue mask. At inference, masks are binarized at 0.5, and the pocket score is the classification probability multiplied by the mean mask probability within the predicted mask—a calibration factor that downweights diffuse masks.

Matching and losses. Hungarian matching between the N_q=50 proposals and the ground-truth pockets minimizes $\mathcal{C} = 2\,\mathcal{C}_{\mathrm{cls}} + 5\,\mathcal{C}_{\mathrm{bce}} + 5\,\mathcal{C}_{\mathrm{dice}}$, where $\mathcal{C}_{\mathrm{cls}}$ is the negative predicted probability of the ground-truth class, $\mathcal{C}_{\mathrm{bce}}$ the per-residue binary cross-entropy between the predicted and ground-truth masks, and $\mathcal{C}_{\mathrm{dice}}$ the Dice discrepancy. The training loss sums the same three terms over matched pairs (unmatched queries contribute only the classification term toward the no-pocket class), normalized by the number of ground-truth pockets, with auxiliary supervision at every decoder layer; the PPN residue-level loss (weight 2) is added on top.

Contrastive denoising (CDN). Each ground-truth pocket spawns denoising queries whose content is its mask-pooled residue feature, $\mathbf{c} = \sum_i w_i\, \mathbf{x}_{\mathrm{res},i}$ with w_i the ground-truth mask normalized over mapped residues, and whose anchor is the masked C_α centroid $\sum_i w_i\, \mathbf{a}_i$. Two positive copies—one clean, one with Gaussian feature noise 0.5 and coordinate noise σ=2 Å—reconstruct their source pocket, while one *contrastive negative* copy with larger noise (feature noise $\times 2$, σ=6 Å) is labeled no-object; the number of DN queries is capped at 32. DN queries attend to the same memory but are attention-isolated from the real queries, skip Hungarian matching (each is aligned 1:1 with its source pocket), and incur mask losses on the positive copies only.

Relation to DETR-style detectors. Hyper-Fold-Pocket borrows the set-prediction formulation of DETR (Carion et al., 2020) but redesigns every component for three-dimensional structures. (i) *No transformer encoder*: DETR-family detectors interpose a transformer encoder between the convolutional backbone and the decoder; we replace it with a lightweight projection neck over the two backbone levels, so the decoder cross-attends directly to a convolutional, geometry-aware memory. (ii) *Structure-anchored queries*: DETR's queries are freely learned embeddings, and anchor-based variants (Liu et al., 2022; Zhang et al., 2023a; Lv et al., 2024) parameterize or select 2-D anchor boxes from encoder outputs; our proposals are anchored on actual residues—content initialized from the top-pocketness residue features, position from their C_α coordinates—fixing both the query content and its 3-D reference point without any encoder. (iii) *Masks instead of boxes*: each proposal predicts a per-residue mask by inner product with the residue keys, rather than regressing a bounding box. (iv) *Contrastive denoising*: DN-DETR and DINO (Li et al., 2022; Zhang et al., 2023a) reconstruct noised ground truth as an auxiliary positive-only task; we additionally label heavily noised copies as *negatives*, teaching the decoder to reject near-miss decoys around real pockets. The bi-level memory plays the role of the multi-scale encoder features of Deformable-DETR (Zhu et al., 2021), but comes from two depths of a single convolutional backbone rather than an image pyramid.

Relation to UniSite (Fan et al., 2025). UniSite-1D/3D instantiate the DETR template almost verbatim for pocket detection: a 6-layer transformer encoder processes fused ESM–GearNet-Edge residue features, and a 6-layer decoder refines 32 *freely learned* site queries (zero-initialized content with learned positional embeddings), read out by classification and inner-product mask heads and trained with Hungarian-matched BCE and Dice losses only. Hyper-Fold-Pocket shares the set-prediction formulation but differs in three ways. (i) *Proposals are discovered, not learned*: our queries are the top-N_q residues by pocketness, carrying the selecting residues' own features as content and their C_α coordinates as 3D-PE anchors, whereas UniSite's queries are data-independent parameters that must localize pockets through cross-attention alone. (ii) *No transformer encoder*: UniSite interposes six encoder layers between the convolutional backbone and the decoder; our decoder acts directly on the bi-level convolutional memory, in which sequence–geometry interactions are already encoded by the backbone. (iii) *Contrastive denoising*: UniSite trains with matched losses only, while CDN adds positive reconstructions and near-miss negatives ($+1.5$ AP@0.5, Figure 7). A fourth, orthogonal difference is that Hyper-Fold-Pocket is purely structural: its residue features come from Hyper-Fold without any sequence model, whereas UniSite-3D leans on ESM features fused with GearNet-Edge.

C EXPERIMENTAL SETUP

Classification baselines: provenance and deviations. GAT, Point Transformer, SchNet, GVP and ProNet are our reproductions under the same recipe (ProNet at backbone level, the variant its authors found best for function prediction); the GCN efficiency numbers are measured on a reference build mirroring the GAT configuration, as Zhang et al. release no official GCN recipe. Point Transformer uses a stabilized recipe (lr 5×10^{-5}, gradient clip 1.0, no AMP), as the unified protocol diverges for this model. ProNet-Backbone (Wang et al., 2023) and SCHull (Wang et al., 2025) modify the input *graph construction*, an axis orthogonal to the layer operator we study; their efficiency numbers are measured on the official SCHull4Science configuration (backbone level, three blocks of width 256)—SCHull adds no parameters, and its hull construction is a preprocessing step. CDConv is the official model trained with its published fold-specific recipe (SGD, 400 epochs, geometry parameter l=11); the EC numbers of CDConv and Hyper-Fold are from our unified-protocol runs, while their fold numbers use the fold-specific recipes.

EC number prediction. All EC models are trained with AdamW (learning rate 10^{-4}, no weight decay), using a 10-epoch linear warmup followed by cosine decay, a batch size of 16, and 300 epochs in total; we apply EMA with decay 0.999, mixed precision, and gradient clipping at norm 10.

Fold classification. The same AdamW recipe and 300-epoch schedule as EC. For Hyper-Fold and Hyper-Fold-Deep we additionally use an SGD recipe with EMA, label smoothing, and structure-forcing augmentation (coordinate noise σ=0.15 Å, residue-type masking p=0.15); Hyper-Fold-Deep uses the widening channel pyramid described in Appendix B, with the narrower sequence window ℓ=5.

Pocket detection. Hyper-Fold-Pocket and all re-trained pocket baselines are optimized with AdamW (learning rate 2×10^{-4}, weight decay 0.05 applied only to parameters with at least two dimensions), using a 3-epoch warmup followed by cosine decay, a batch size of 16, and 40 epochs in total; we apply EMA with decay 0.999, clip gradients at norm 0.5, and add Gaussian coordinate noise (σ=0.2 Å) to the Cα positions during training.

Latency measurement. All models are trained and benchmarked on a single A100-80GB; latency is measured at batch size 1 on a synthetic length-300 protein (10 warmup and 50 timed forward passes, fp32), excluding PDB parsing and ESM tokenization. Graph construction follows each model's native pipeline: it is part of the timed forward pass for models that build graphs on-device (Hyper-Fold, UniSite, ProNet, CDConv), and is performed once outside the timed region for models whose framework treats the graph as a preprocessing artifact (torchdrug-based GCN, GAT and GVP). For the pocket baselines of Table 2: UniSite-1D / UniSite-3D are 662.9M / 683.9M parameters at 46.4 / 114.3 ms (including the ESM-2 embedding stage, which alone costs 38.1 ms); DeepPocket is timed as its two-stage pipeline (a 25.9M segmentation U-Net at 5.8 ms plus a 0.7M candidate-ranking classifier at 0.6 ms per candidate center); VN-EGNN consumes precomputed ESM features that are not counted in its parameter or latency figures; Fpocket, Fpocket-rescore and P2Rank are geometry-based CPU tools with no neural parameters and are not timed on GPU.

D MORE EVALUATION ON THE HYPER-FOLD BACKBONE

D.1 FULL EC RESULTS

Table 4 reports the full EC benchmark behind the two summary columns of Table 1: test Fmax at five sequence-identity cutoffs, together with micro-AUPR at the 95% cutoff. Hyper-Fold is the best method at every cutoff, and its advantage over the strongest baseline widens as the cutoff tightens, from $+1.1$ Fmax points at 95% (0.864 vs. 0.853) to $+2.1$ points at 30% (0.726 vs. 0.705)—the gain concentrates on the hard, remote-homology cases where geometry must substitute for sequence similarity. Hyper-Fold-Deep trails Hyper-Fold slightly but stays ahead of every baseline at all five cutoffs.

Table 4: EC number prediction: test Fmax at five sequence-identity cutoffs (30/40/50/70/95%) and micro-AUPR at the 95% cutoff. Method order, provenance, and the Params/Latency measurements are as in Table 1. **Bold**: best overall; underlined: second best overall.

			Fmax at sequence-identity cutoff					AUPR
Method	Params	Latency	30%	40%	50%	70%	95%	95%
Generic architectures								
GCN	21.9M	6.2 ms	0.245	0.246	0.246	0.280	0.320	0.319
GAT	21.9M	6.9 ms	0.246	0.248	0.246	0.276	0.329	0.339
Point Transformer	14.5M	9.7 ms	0.247	0.260	0.276	0.320	0.377	0.346
SchNet	6.8M	8.1 ms	0.555	0.573	0.617	0.686	0.742	0.714
Protein-specific architectures								
GVP	7.1M	10.0 ms	0.347	0.338	0.347	0.400	0.480	0.467
GearNet	31.1M	4.0 ms	0.557	0.570	0.615	0.693	0.730	0.751
GearNet-Edge	40.7M	48.6 ms	0.625	0.646	0.694	0.757	0.810	0.835
GearNet-Edge-IEConv	46.4M	44.1 ms	0.644	0.674	0.718	0.774	0.813	0.831
ProNet	1.5M	9.1 ms	0.582	0.621	0.669	0.726	0.765	0.795
ProNet-Backbone	4.3M	11.9 ms	0.537	0.577	0.629	0.694	0.737	0.755
CDConv	31.0M	14.2 ms	0.705	0.737	0.777	0.823	0.853	0.859
SCHull	4.3M	11.9 ms	0.562	0.608	0.664	0.729	0.771	0.787
Ours								
Hyper-Fold	7.7M	10.5 ms	**0.726**	**0.752**	**0.789**	**0.832**	**0.864**	**0.874**
Hyper-Fold-Deep	20.6M	15.7 ms	0.716	0.746	0.787	0.830	0.861	0.873

D.2 Architecture and geometry ablations

Figure 5 ablates the backbone's geometry schedule, sequence window, and width on the same benchmark (configurations in Table 5). All runs use the identical 300-epoch recipe (AdamW 10^{-4}, cosine schedule, EMA 0.999) as the EC main table; numbers are test Fmax in percent at five sequence-identity cutoffs.

Table 5: Configurations of the EC ablation variants visualized in Figure 5: the per-layer radius schedule r (in Å) and the sequence window ℓ. All rows share the six-layer Hyper-Fold recipe (6.67M parameters, bottleneck $C/2$) and the unified 300-epoch protocol; only the stated schedules vary. Cell shading encodes magnitude within each schedule (darker = larger).

		r (Å)						ℓ					
Group	Variant	L1	L2	L3	L4	L5	L6	L1	L2	L3	L4	L5	L6
Radii	Uniform Radii	10	10	10	10	10	10	21	21	21	21	21	21
	Expanding Radii (8→12 Å)	8	8	10	10	12	12	17	17	17	21	21	21
	Contracted Radii (6→12 Å)	6	6	8	8	10	12	13	13	17	17	21	21
Window	Uniform Window	8	8	8	10	10	10	21	21	21	21	21	21
	Contracted + Long Window	6	6	8	10	12	12	13	13	17	17	21	25
Width	Quarter Width ($C/4$)	8	8	8	10	10	10	17	17	17	21	21	21
Default	Hyper-Fold	8	8	8	10	10	10	17	17	17	21	21	21

Three findings emerge from Figure 5. First, the shallow-layer geometry matters more than the deep-layer one: shrinking the shallow radii from 10 Å to 8 Å is free (Uniform Radii vs. default: 78.55 vs. 78.87 Fmax@50), and reducing them further to 6 Å is harmful (Contracted Radii)—8 Å is the shallow-layer sweet spot, consistent with the local chemistry of residue neighborhoods. Second,

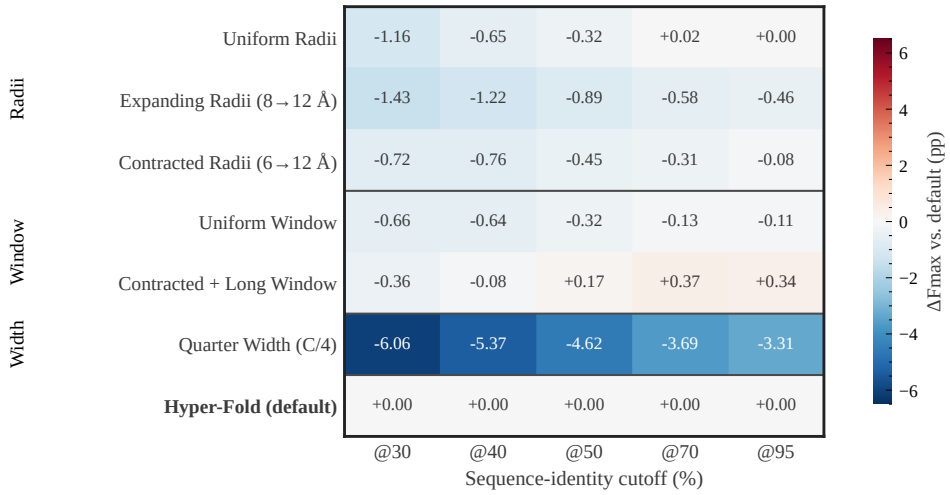


Figure 5: EC architecture and geometry ablations as a ΔFmax heatmap: each cell is the variant's test Fmax minus the default Hyper-Fold configuration's at the same sequence-identity cutoff (configurations in Table 5). Rows are grouped by ablated axis; only the quarter-width variant departs markedly from the default.

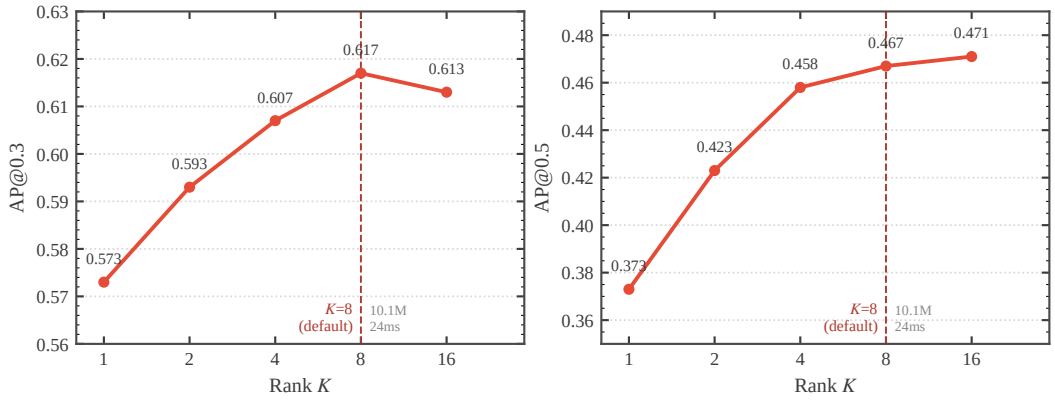


Figure 6: Rank-K saturation of Hyper-Fold-Pocket: AP@0.3 (left) and AP@0.5 (right) of the rank sweep, plotted separately because the two metrics live on different scales. Accuracy saturates at $K{=}8$; parameters and latency are annotated at the working point. The operator spectrum that explains the saturation is in Figure 9.

width is the binding constraint: a quarter-width model collapses by 4.6 Fmax@50 points (74.25 vs. 78.87), while the bottleneck $C/2$ design matches full width, so expressivity should be bought with the rank-K kernel rather than raw channels. Third, the window and radius schedules interact: combining contracted shallow radii with a longer deep-layer window (Contracted + Long Window) trades strict-cutoff accuracy for lenient-cutoff gains, whereas the default schedule is the strongest at the strict 30/40% cutoffs—the remote-homology regime where geometry must substitute for sequence similarity.

E More evaluation on Hyper-Fold-Pocket

E.1 Pocket ablations

Three ablation axes are isolated under the identical head and training recipe: the gating rank K (Figure 6), the readout (Figure 8), and the head components (Figure 7). The aggregation-operator ladder under the same shared backbone is in Table 3.

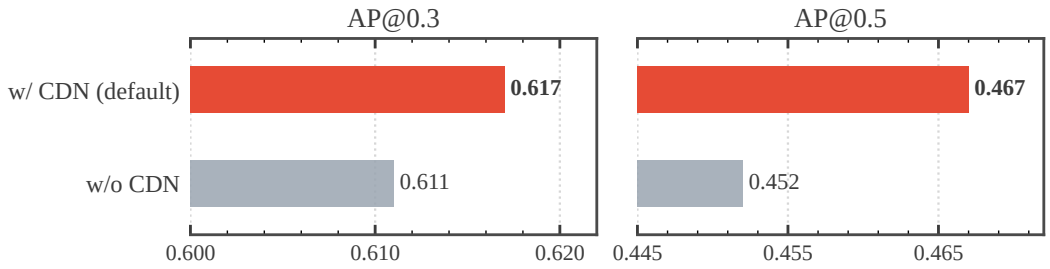


Figure 7: Head-component ablation, split by metric (left: AP@0.3, right: AP@0.5). Coral bars mark the default configuration: removing CDN costs 0.6/1.5 points (AP@0.3/AP@0.5) at identical latency.

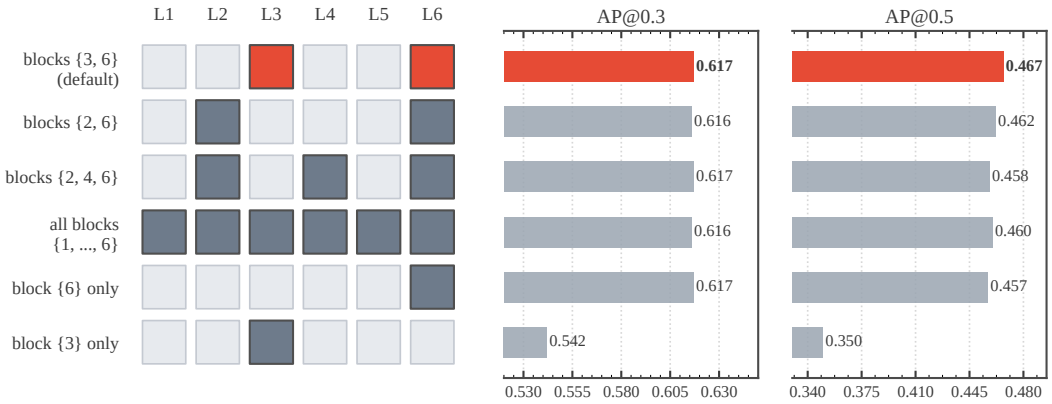


Figure 8: Readout selection of Hyper-Fold-Pocket. Left: which block outputs feed the detection neck (coral: selected). Right: AP@0.3 and AP@0.5 of the readout ablation. Multi-scale readouts tie within noise on AP@0.3; the single-scale $\{3\}$-only readout collapses and $\{6\}$-only trails on AP@0.5, so we keep the cheapest two-scale configuration $\{3, 6\}$.

Rank K. AP rises monotonically from $K{=}1$ to $K{=}8$ ($+4.4$ AP@0.3, $+9.4$ AP@0.5) and saturates: $K{=}16$ adds nothing on AP@0.3 and 0.4 points on AP@0.5 despite 31% more parameters and 25% higher latency. The saturation point coincides with the operator spectrum measurement (Appendix F.6), where the learned kernel is dominated by its leading components, so additional ranks fit noise. Note that the strict-IoU column gains twice as much as the lenient one ($+9.4$ vs. $+4.4$): richer gating buys precise pocket delineation, not coarse localization.

Readout. The two-scale $\{3, 6\}$ readout is robust (Figure 8): every multi-scale tap—$\{2, 6\}$ (0.616/0.462), $\{2, 4, 6\}$ (0.617/0.458), and all six blocks (0.616/0.460)—is within noise of the default on AP@0.3, while the default keeps the best AP@0.5 at the lowest cost. Single-scale readouts expose the division of labor: $\{6\}$-only matches AP@0.3 (0.617) but drops 1.0 point on AP@0.5, whereas $\{3\}$-only collapses to 0.542/0.350—deep features suffice for coarse localization, but precise delineation needs the shallow tap.

Contrastive denoising. Removing CDN costs 0.6 AP@0.3 and 1.5 AP@0.5 at identical latency. That the stricter threshold again benefits more matches CDN's role: the contrastive negatives teach the decoder to reject near-miss decoys, which is exactly what a high-IoU evaluation penalizes.

Across all three groups, design choices move AP@0.5 about twice as much as AP@0.3—the strict-IoU column is where content–geometry binding pays off.

E.2 Qualitative analysis protocols (Figure 4)

Predictions and matching. All panels are rendered from a single dump of the released Hyper-Fold-Pocket checkpoint (UniSite-DS test split, 2,293 proteins; sequences longer than 800 residues

excluded at inference). Predicted masks are binarized at a mask-logit sigmoid of 0.5; the query score is the product of its classification probability and the mean mask probability over the predicted region. Matching follows the AP evaluation exactly: per protein, queries sorted by descending score greedily claim their best-IoU ground-truth pocket, and a query is a true positive iff its best IoU reaches the threshold and that pocket is not already covered; each ground-truth pocket is claimable at most once. IoU is computed on residue sets (boolean intersection over union of the index sets). The UniSite-3D comparison uses its official per-protein predictions on the same split, scored by the identical matching code and ground truth.

(A) Error maps. Two in-distribution test proteins (Q57849, P00359), three columns (ground truth, UniSite-3D, Hyper-Fold-Pocket). For Hyper-Fold-Pocket we show, among the top-5 queries, the prediction with the highest IoU against the union of ground-truth pocket masks; for UniSite-3D we show its highest-scoring prediction. Residues are colored by set operations on this pair of masks: TP $=$ pred $\cap$ GT (green), FP $=$ pred $\setminus$ GT (blue), FN $=$ GT $\setminus$ pred (red); the ground-truth column shows the pocket in purple. The annotation under each prediction is the residue-set IoU of the shown mask against the ground-truth union. All views are ray-traced molecular surfaces rendered with PyMOL, oriented so that the pocket opening faces the camera (viewing direction from the protein C_α centroid to the pocket C_α centroid), with a 10 Å scale bar calibrated by two pseudoatoms.

(B) Pocket anatomy. The P00359 pocket with its bound NAD ligand (sticks; C yellow, O red, N blue, P orange), the whole protein as a white semi-transparent surface with the ground-truth pocket tinted purple. The close-up magnifies the dashed region—anchored on the ligand—with the same TP/FP/FN error coloring as (A).

(C) Query constellation. All 50 decoder queries of a single forward pass on P00359, rendered as spheres on the molecular surface. Each sphere sits at the C_α centroid of the query's predicted mask (queries with an empty mask are not drawn); the radius is 1.2 Å, enlarged to 2.0 Å when the query score exceeds 0.5. Sphere color encodes the PPN pocketness score on a linear grey-blue $\rightarrow$ vermillion scale. The purple patch is the top-1 query's predicted mask.

(D) PR curves. Precision–recall curves are *pooled*: the (score, TP/FP) records of all queries of all test proteins are concatenated, sorted globally by descending score, and accumulated; recall is relative to the total number of ground-truth pockets in the split. AP is the area under the precision envelope (trapezoidal). Curves use all 50 queries per protein at IoU 0.3 and 0.5; their areas are the AP numbers of Table 2. The UniSite-3D curve is produced by the same pooling pipeline applied to its official predictions.

(E) Score histogram. Distribution of the top-5 query scores per protein over the same split (40 equal-width bins on $[0, 1]$), split into true and false positives at IoU 0.5 by the matching rule above. Note the different query budget from (D): the histogram reflects the top-5 operating point, whereas the PR curves integrate over all 50 queries.

F Proofs

F.1 Proof of Proposition 1 (containment)

Proof. Channel-gated $\subset$ matrix-gated. Take the diagonal matrix-valued kernel $\mathbf{W}(\boldsymbol{\delta}) = \mathrm{diag}(\mathbf{w}(\boldsymbol{\delta}))$; then $\mathbf{W}(\boldsymbol{\delta})\mathbf{x} = \mathbf{w}(\boldsymbol{\delta}) \odot \mathbf{x}$ is exactly class (3).

Scalar-gated $\subset$ channel-gated. Absorb the fixed linear map $\mathbf{V}$ into the content features, $\tilde{\mathbf{x}}_j = \mathbf{V}\mathbf{x}_j$, and take the channel gate $\mathbf{w}(\boldsymbol{\delta}) = \alpha(\boldsymbol{\delta})\,\mathbf{1}$ with all entries tied. Then $\mathbf{w}(\boldsymbol{\delta}) \odot \tilde{\mathbf{x}}_j = \alpha(\boldsymbol{\delta})\,\mathbf{V}\mathbf{x}_j$ is exactly class (2).

Additive MP $\subset$ matrix-gated. Augment the content with a constant channel, $\tilde{\mathbf{x}} = [\mathbf{x}; 1] \in \mathbb{R}^{C+1}$, and define the geometry-affine kernel $\mathbf{W}(\boldsymbol{\delta}) = [\,\mathbf{W} \mid \mathbf{U}\boldsymbol{\delta}\,] = \mathbf{B}_0 + \sum_{m=1}^{d_g} \delta_m\, \mathbf{B}_m$ with $\mathbf{B}_0 = [\mathbf{W} \mid 0]$ and $\mathbf{B}_m = [\,0 \mid \mathbf{U}\mathbf{e}_m\,]$. Then $\mathbf{W}(\boldsymbol{\delta}_{ij})\tilde{\mathbf{x}}_j = \mathbf{W}\mathbf{x}_j + \mathbf{U}\boldsymbol{\delta}_{ij}$ is exactly class (1): additive MP is the matrix-gated layer whose kernel is affine in $\boldsymbol{\delta}$ (first-order in geometry, no learned geometry-dependent modulation beyond the linear term).

Matrix-gated ⊂ complete bilinear. Fix a basis $\{\mathbf{B}_c\}$ of the matrix space containing the kernel's range and write $\mathbf{W}(\boldsymbol{\delta}) = \sum_c g_c(\boldsymbol{\delta})\,\mathbf{B}_c$ (any continuous kernel admits such an expansion; for a universal ϕ whose coordinates include the coefficient functions $g_1, \dots, g_K$, e.g. a Schauder basis of the continuous functions on the compact edge domain, every continuous kernel is represented exactly). Define the linear map $\mathcal{T}$ on $\mathbb{R}^C \otimes \mathbb{R}^{d_\phi}$ by $\mathcal{T}(\mathbf{x} \otimes \mathbf{e}_c) = \mathbf{B}_c\mathbf{x}$ on the basis tensors, extended linearly. Then

$$\mathcal{T}\big(\mathbf{x} \otimes \phi(\boldsymbol{\delta})\big) = \sum_c g_c(\boldsymbol{\delta})\,\mathcal{T}(\mathbf{x} \otimes \mathbf{e}_c) = \sum_c g_c(\boldsymbol{\delta})\,\mathbf{B}_c\mathbf{x} = \mathbf{W}(\boldsymbol{\delta})\,\mathbf{x},$$

which is class (4). □

F.2 Proof of Proposition 2 (binding blindness)

Proof. For an additive layer (1), both configurations give

$$\mathbf{h}_i^A = \mathbf{W}\mathbf{x}_1 + \mathbf{W}\mathbf{x}_2 + \mathbf{U}\boldsymbol{\delta}_1 + \mathbf{U}\boldsymbol{\delta}_2 = \mathbf{h}_i^B,$$

since the content sum and the geometry sum factorize and addition is commutative; no parameter choice can make $\mathbf{h}_i^A \neq \mathbf{h}_i^B$.

For a matrix-gated layer, the output difference is

$$\mathbf{h}_i^A - \mathbf{h}_i^B = \mathbf{W}(\boldsymbol{\delta}_1)\mathbf{x}_1 + \mathbf{W}(\boldsymbol{\delta}_2)\mathbf{x}_2 - \mathbf{W}(\boldsymbol{\delta}_2)\mathbf{x}_1 - \mathbf{W}(\boldsymbol{\delta}_1)\mathbf{x}_2 = \big(\mathbf{W}(\boldsymbol{\delta}_1) - \mathbf{W}(\boldsymbol{\delta}_2)\big)(\mathbf{x}_1 - \mathbf{x}_2),$$

which is nonzero whenever $\mathbf{W}(\boldsymbol{\delta}_1)(\mathbf{x}_1 - \mathbf{x}_2) \neq \mathbf{W}(\boldsymbol{\delta}_2)(\mathbf{x}_1 - \mathbf{x}_2)$. Such kernels exist: since $\boldsymbol{\delta}_1 \neq \boldsymbol{\delta}_2$, a continuous matrix-valued kernel may take $\mathbf{W}(\boldsymbol{\delta}_1) = I$ and $\mathbf{W}(\boldsymbol{\delta}_2) = 0$ (and universal weight networks approximate it), giving $\mathbf{h}_i^A = \mathbf{x}_1 \neq \mathbf{x}_2 = \mathbf{h}_i^B$. □

F.3 Proof of Theorem 1 (the bilinear ceiling)

Proof. Let the message $m(\mathbf{x}_j, \boldsymbol{\delta}_{ij})$ be second-order, i.e. bilinear in $(\mathbf{x}_j, \phi(\boldsymbol{\delta}_{ij}))$. By the universal property of the tensor product, every bilinear map on $\mathbb{R}^C \times \mathbb{R}^{d_\phi}$ factors uniquely through $\mathbf{x}_j \otimes \phi(\boldsymbol{\delta}_{ij})$: there exists a linear $\mathcal{T}$ with $m(\mathbf{x}_j, \boldsymbol{\delta}_{ij}) = \mathcal{T}\big(\mathbf{x}_j \otimes \phi(\boldsymbol{\delta}_{ij})\big)$. Hence the message depends on $(\mathbf{x}_j, \boldsymbol{\delta}_{ij})$ only through their outer product, which is therefore a sufficient statistic of all second-order interaction. Conversely, every linear $\mathcal{T}$ on $\mathbb{R}^C \otimes \mathbb{R}^{d_\phi}$ defines a message $\mathcal{T}(\mathbf{x}_j \otimes \phi(\boldsymbol{\delta}_{ij}))$ that is bilinear in $(\mathbf{x}_j, \phi(\boldsymbol{\delta}_{ij}))$, so class (5) realizes *all* such messages. Neighborhood summation and layer stacking preserve the class by linearity, so no composition of such layers exceeds this ceiling. □

F.4 Exactness of the outer-product factorization

Lemma 1 (Outer product realizes matrix gating)**.** *Let $P_c : \mathbb{R}^{K\times C} \to \mathbb{R}^{C'}$ be linear, identified with the block matrix $P_c = [\,P_c^{(1)} \mid \cdots \mid P_c^{(K)}\,]$, $P_c^{(k)} \in \mathbb{R}^{C'\times C}$, acting on the row-major flattening of $\mathbb{R}^{K\times C}$. Then for every $\mathbf{g} \in \mathbb{R}^K$ and $\mathbf{h} \in \mathbb{R}^C$,*

$$P_c\big(\mathbf{g} \otimes \mathbf{h}\big) = \Big(\sum_{k=1}^{K} g_k\, P_c^{(k)}\Big)\mathbf{h}. \tag{11}$$

Consequently, with $\mathbf{h}_j = \mathbf{W}_v\mathbf{x}_j$ and $\mathbf{B}_k = P_c^{(k)}\mathbf{W}_v$, the layer of Eq. (8) coincides exactly with the matrix-gated form of Eq. (9) with kernel $\mathbf{W}(\boldsymbol{\delta}, \Delta s) = \sum_{k=1}^K \tilde{g}_k(\boldsymbol{\delta}, \Delta s)\,\mathbf{B}_k$, $\tilde{g}_k = \sigma\, g_k$.

Proof. The row-major flattening of $\mathbf{g}\otimes\mathbf{h}$ is the concatenation of the blocks $g_k\mathbf{h} \in \mathbb{R}^C$, $k = 1, \dots, K$. Block-wise multiplication gives $P_c(\mathbf{g} \otimes \mathbf{h}) = \sum_k P_c^{(k)}(g_k\mathbf{h}) = \sum_k g_k(P_c^{(k)}\mathbf{h}) = \big(\sum_k g_k P_c^{(k)}\big)\mathbf{h}$, where the second equality is linearity of $P_c^{(k)}$ in the face of the scalar g_k. Applying this edge-wise with $g_k = \sigma_{ij}\, g_k(\boldsymbol{\delta}_{ij}, \Delta s_{ij})$, $\mathbf{h}_j = \mathbf{W}_v\mathbf{x}_j$, and summing over $j \in \mathcal{N}(i)$ yields Eq. (9) with $\mathbf{B}_k = P_c^{(k)}\mathbf{W}_v$. □

F.5 A SPECTRAL TAIL BOUND FOR RANK-K GATING

We quantify the error of restricting the edge-conditioned kernel to a rank-K basis. Let the edge descriptor $(\boldsymbol{\delta}, \Delta s)$ range over its data-induced distribution and let $\mathbf{W}(\boldsymbol{\delta}, \Delta s) \in \mathbb{R}^{C' \times C}$ denote the learned full operator family, e.g. the operators induced by the $K{=}16$ model measured in Appendix F.6. Write $\mathbf{w} = \mathrm{vec}(\mathbf{W}) \in \mathbb{R}^{C'C}$ and let the second-moment matrix $M = \mathbb{E}[\mathbf{w}\mathbf{w}^\top] \succeq 0$ have eigenvalues $\lambda_1 \geq \cdots \geq \lambda_{C'C}$ with orthonormal eigenvectors $\mathbf{v}_c$; set $\mathbf{V}_c = \mathrm{mat}(\mathbf{v}_c) \in \mathbb{R}^{C' \times C}$.

Proposition 4 (Spectral tail bound)**.** *For every choice of K basis operators $\{\mathbf{B}_c\}_{c=1}^K \subset \mathbb{R}^{C' \times C}$ and every choice of measurable coefficient functions $g_c(\boldsymbol{\delta}, \Delta s)$,*

$$\mathbb{E}\left\| \mathbf{W}(\boldsymbol{\delta}, \Delta s) - \sum_{c=1}^{K} g_c(\boldsymbol{\delta}, \Delta s)\, \mathbf{B}_c \right\|_F^2 \;\geq\; \sum_{c>K} \lambda_c. \tag{12}$$

Equality holds for the eigenbasis $\mathbf{B}_c = \mathbf{V}_c$ with the orthogonal-projection coefficients $g_c(\boldsymbol{\delta}, \Delta s) = \langle \mathbf{W}(\boldsymbol{\delta}, \Delta s), \mathbf{V}_c \rangle_F$.

Proof. The best-approximation error depends on $\{\mathbf{B}_c\}$ only through $\mathrm{span}\{\mathbf{B}_c\}$, so by Gram–Schmidt we may take the $\mathbf{B}_c$ orthonormal with respect to the Frobenius inner product. For a fixed basis the error is minimized pointwise in $(\boldsymbol{\delta}, \Delta s)$ by the orthogonal projection $g_c = \langle \mathbf{W}, \mathbf{B}_c \rangle_F$, and Pythagoras gives, with $\mathbf{b}_c = \mathrm{vec}(\mathbf{B}_c)$,

$$\mathbb{E}\left\| \mathbf{W} - \sum_{c \leq K} g_c \mathbf{B}_c \right\|_F^2 = \mathbb{E}\,\|\mathbf{w}\|_2^2 - \sum_{c=1}^{K} \mathbb{E}\,\langle \mathbf{w}, \mathbf{b}_c \rangle^2 = \mathrm{tr}(M) - \sum_{c=1}^{K} \mathbf{b}_c^\top M\, \mathbf{b}_c.$$

By Ky Fan's maximum principle, $\max \sum_{c=1}^K \mathbf{b}_c^\top M \mathbf{b}_c = \sum_{c=1}^K \lambda_c$ over orthonormal sets $\{\mathbf{b}_c\}$, attained by the top-K eigenvectors. The minimum error is therefore $\mathrm{tr}(M) - \sum_{c \leq K} \lambda_c = \sum_{c>K} \lambda_c$, attained by $\mathbf{B}_c = \mathbf{V}_c$. □

Corollary 1 (Relative error of the deployed model)**.** *The relative operator error of the optimal rank-K factorization is the spectral tail fraction, $\mathbb{E}\|\mathbf{W} - \widehat{\mathbf{W}}_K\|_F^2 / \mathbb{E}\|\mathbf{W}\|_F^2 = 1 - \big(\sum_{c \leq K} \lambda_c\big)/\mathrm{tr}(M)$. With the top-8 energy fractions measured in Figure 9 (89.9–98.2% per block), rank-8 gating incurs a relative operator error of at most 10.1% in every backbone block, and at most 8.2% in blocks 1–3.*

Corollary 2 (Error propagation through the layer)**.** *Let $\widehat{\mathbf{W}}$ be any rank-K approximation of $\mathbf{W}$ and write $\Delta\mathbf{W}_{ij} = \mathbf{W}(\boldsymbol{\delta}_{ij}, \Delta s_{ij}) - \widehat{\mathbf{W}}(\boldsymbol{\delta}_{ij}, \Delta s_{ij})$. Then the outputs of Eq.* (9) *satisfy*

$$\|\mathbf{h}_i - \widehat{\mathbf{h}}_i\|_2^2 \;\leq\; \Big(\sum_{j \in \mathcal{N}(i)} \|\Delta\mathbf{W}_{ij}\|_F^2 \Big) \Big(\sum_{j \in \mathcal{N}(i)} \|\mathbf{W}_v \mathbf{x}_j\|_2^2 \Big).$$

Proof. $\mathbf{h}_i - \widehat{\mathbf{h}}_i = \sum_j \Delta\mathbf{W}_{ij}\, \mathbf{W}_v \mathbf{x}_j$; apply the triangle inequality, the bound $\|\Delta\mathbf{W}_{ij}\mathbf{W}_v\mathbf{x}_j\|_2 \leq \|\Delta\mathbf{W}_{ij}\|_F\, \|\mathbf{W}_v\mathbf{x}_j\|_2$, and Cauchy–Schwarz. □

Remark 1 (The tail bound is a floor, not the achieved error)**.** *Eq.* (12) *lower-bounds the error of* every *rank-K scheme whose coefficients are arbitrary functions of the edge descriptor. Hyper-Fold's gates are further constrained to the hypothesis class of the bucketed weight network (Eq.* (7)*), so the achieved error lies above the spectral tail; the saturation of the K-sweep in Figure 6 indicates that at $K{=}8$ the residual error is dominated by this realizability gap rather than by the rank.*

Remark 2 (The realizability gap vanishes with capacity)**.** *The edge descriptor $(\boldsymbol{\delta}, \Delta s)$ ranges over a compact domain, the bucket map is piecewise constant, and within each bucket the weight network is an MLP with LeakyReLU activations—a universal approximator of continuous functions. Any measurable coefficient map $g_c(\boldsymbol{\delta}, \Delta s)$ can therefore be approximated uniformly to arbitrary precision as the hidden width grows, so the gap between the achieved rank-K error and the spectral floor of Proposition 4 is a capacity effect, not a structural limit of the factorization.*

Remark 3 (Sequence clamping)**.** *The discretization $\mathrm{bucket}(\Delta s)$ is exact for $|\Delta s| < \ell/2$ (each integer offset owns a bucket) and merges all pairs with $|\Delta s| \geq \ell/2$ into the two boundary buckets. The smooth gate σ_{ij} is decreasing in $\bar{s} = |\Delta s_{ij}|/(\ell/2)$ and vanishes as $\bar{d}\,\bar{s} \to 1$, so the merged pairs—those maximal in sequence distance—are precisely the ones whose contribution is softly removed, which attenuates the clamping error in the layer output.*

F.6 Operator spectrum measurement

To substantiate Proposition 3 we measure the spectrum of the learned full operators of a $K{=}16$ Hyper-Fold-Pocket model—deliberately over-parameterized relative to the deployed $K{=}8$, so the measurement reveals how many gating directions the model actually uses. For each block we collect the gate coefficients $g(\boldsymbol{\delta}_{ij}) \in \mathbb{R}^K$ over all edges of 64 validation structures into the second-moment matrix $G = \mathbb{E}[gg^\top]$; with the Gram matrix of the basis matrices, $\mathrm{Gram}_{cc'} = \langle \mathbf{B}_c, \mathbf{B}_{c'} \rangle_F$, the operator energy is $\mathbb{E}\|\mathbf{W}\|_F^2 = \mathrm{tr}(G\,\mathrm{Gram})$, and diagonalizing $G^{1/2}\,\mathrm{Gram}\,G^{1/2}$ yields the operator spectrum. We report the fraction of its total eigenvalue energy captured by the top-8 eigendirections (Figure 9).

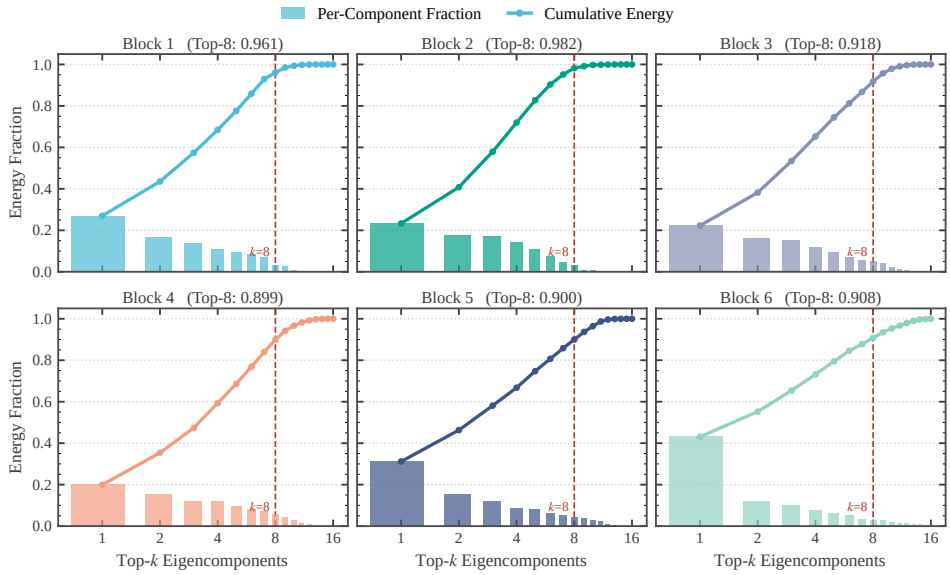


Figure 9: Per-block operator spectrum of the $K{=}16$ Hyper-Fold-Pocket model. Light bars are the individual eigenvalue-energy fractions; the colored line is the cumulative energy. The dashed line marks the deployed rank $K{=}8$; panel titles report the top-8 fraction. Every block's spectrum decays rapidly, and deep blocks are no flatter than shallow ones.

Consistent with this measurement, the K-sweep on pocket detection (Figure 6) shows that doubling the basis from $K{=}8$ to 16 (a 31% parameter cost) yields no AP@0.3 gain and only $+0.4$ AP@0.5 points—accuracy saturates where the spectrum does.