LEARNING WHEN TO UPDATE: DYNAMIC MASKING FOR ADAPTIVE REPRESENTATION REFINEMENT

Anonymous Authors

ABSTRACT

Deep neural networks typically incorporate residual updates with a fixed coefficient even though the computational role of individual content representations can vary across inputs and depth. We introduce Dynamic Masking (DYM), a parameter-efficient mechanism that learns input-dependent coefficients controlling how strongly newly computed updates enter the residual stream. The learned coefficients participate directly in computation and can be inspected after inference, but are not assumed to be feature-importance scores. We evaluate DYM across structural biology, language modeling, computer vision, and 3D medical image segmentation. In antibody–antigen interface prediction, interface residues undergo systematically stronger attenuation than non-interface residues. In language models, modulation is structured by linguistic category and context, and direct interventions show that strongly modulated tokens are computationally consequential. Inference-time gate ablations further show that full token- and hidden-dimension-dependent modulation cannot be reduced to static residual scaling: collapsing DYM to a token scalar or a fixed branch scalar substantially degrades Pythia-160M performance, while shuffling learned gate vectors across tokens is still more disruptive. In vision and segmentation, signed modulation is spatially structured and task dependent. Across domains, relevant representations may be amplified or attenuated, supporting DYM as a mechanism for adaptive representation refinement rather than a conventional importance estimator.

1 INTRODUCTION

Residual architectures refine representations by repeatedly adding newly computed transformations to an existing state. In Transformers, for example, attention and feed-forward branches are typically incorporated through residual additions with a fixed coefficient of one (Vaswani et al., 2017). Although the update itself is input dependent, the strength with which it modifies the residual representation is usually not adapted to individual content units. This uniform treatment contrasts with the heterogeneous role of features in many tasks: Vision Transformers represent images as patch-token sequences (Dosovitskiy et al., 2020), amino-acid residues play distinct roles in molecular interactions, and the relevance of language tokens varies with context. Prior work has shown that learned modulation can regulate information flow through gating, conditional feature modulation, channel recalibration, or adaptive token selection (Srivastava et al., 2015; Perez et al., 2018; Hu et al., 2018; Ryoo et al., 2021).

A separate literature seeks to identify which features contribute to predictions. Integrated Gradients and Grad-CAM provide post hoc estimates of feature relevance (Sundararajan et al., 2017; Selvaraju et al., 2017), while attention weights are often inspected as model-internal signals. However, post hoc attribution is computed after the predictive mechanism has operated, visually plausible explanations need not be faithful to the underlying computation (Adebayo et al., 2018), and the explanatory role of attention remains debated (Jain and Wallace, 2019; Wiegreffe and Pinter, 2019). We therefore ask a different question: can the network expose an internal control signal that is both part of the forward computation and interpretable as a record of how strongly each representation is being updated?

We introduce Dynamic Masking (DYM), which is inserted between a functional transformation and its residual addition. DYM learns how strongly a newly computed update should modify each content-bearing representation without removing the representation itself or changing sequence length. The learned coefficient has two related roles. Computationally, it directly strengthens or attenuates the candidate update. Analytically, it exposes how update strength is distributed across language

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.

tokens, residues, image patches, or spatial locations. We do *not* define this coefficient as feature importance: a strongly attenuated update can be highly consequential, and the effect of a coefficient also depends on the update it scales. DyM also differs from attention. Attention determines how information is exchanged among representations to construct an update, whereas DyM determines how strongly the resulting update is incorporated into the residual stream.

We evaluate DyM across autoregressive language modeling, antibody–antigen interface prediction, ImageNet classification, and 3D brain-tumor segmentation. The cross-domain experiments are designed to separate predictive performance from interpretation of the modulation signal. Our contributions are fourfold: (1) an architecture-compatible mechanism for content-dependent residual-update strength; (2) analyses that distinguish update regulation from feature importance using external annotations and direct interventions; (3) inference-time gate ablations in Pythia that separate static branch scaling, token-dependent gating, hidden-dimension-specific modulation, and token-to-gate alignment; and (4) evidence across four domains that computationally relevant representations may be either amplified or attenuated rather than following a universal monotonic rule.

2 Related Work

Adaptive information flow. Highway Networks introduced transform and carry gates controlling the balance between transformed and propagated information (Srivastava et al., 2015). FiLM performs conditional feature-wise affine modulation (Perez et al., 2018), while Squeeze-and-Excitation networks learn input-dependent channel recalibration (Hu et al., 2018). TokenLearner extends adaptive computation to visual content units by dynamically selecting informative regions (Ryoo et al., 2021). DyM shares the principle that information flow need not be uniform, but controls a different object: all content units are retained, while the strength of each newly computed residual update is continuously regulated. This makes the same mechanism applicable after attention, feed-forward transformations, graph message passing, or other residual functions.

Attribution, attention, and intrinsic interpretation. Post hoc attribution methods such as Integrated Gradients, Grad-CAM, and SHAP estimate associations between input features and predictions (Sundararajan et al., 2017; Selvaraju et al., 2017; Lundberg and Lee, 2017); their outputs do not themselves control the original forward computation. Faithfulness therefore requires separate evaluation (Adebayo et al., 2018; DeYoung et al., 2020). Attention is internal to computation, but large attention weights need not imply proportionally large causal effects (Jain and Wallace, 2019; Serrano and Smith, 2019), while other work argues that attention can still be explanatory when the claim and evaluation criterion are clearly specified (Wiegreffe and Pinter, 2019). Learned-rationale methods instead make selection part of the predictive computation (Lei et al., 2016; Bastings et al., 2019). DyM is complementary: it neither selects a subset of inputs nor assumes that its coefficient is an importance score. It exposes an adaptive, layer-dependent decision about representation refinement, and the meaning of that decision is established empirically through annotations and interventions.

3 Dynamic Masking

3.1 Dynamic update modulation

For a content-bearing unit i at layer l, a standard residual operation is

$$\mathbf{h}_i^{(l+1)} = \mathbf{h}_i^{(l)} + \Delta\mathbf{h}_i^{(l)}. \tag{1}$$

DyM replaces the fixed update coefficient by an input-dependent modulation term,

$$\mathbf{h}_i^{(l+1)} = \mathbf{h}_i^{(l)} + m_i^{(l)}\Delta\mathbf{h}_i^{(l)}, \tag{2}$$

with the scalar form

$$m_i^{(l)} = 2\sigma\left(\mathbf{w}^{(l)\top}\Delta\mathbf{h}_i^{(l)} + b^{(l)}\right), \qquad 0 < m_i^{(l)} < 2. \tag{3}$$

The identity $m = 1$ recovers the original residual operation. Values below one attenuate the newly computed update and values above one strengthen it. Importantly, DyM does not remove or directly scale the existing residual state; it regulates only newly computed information.

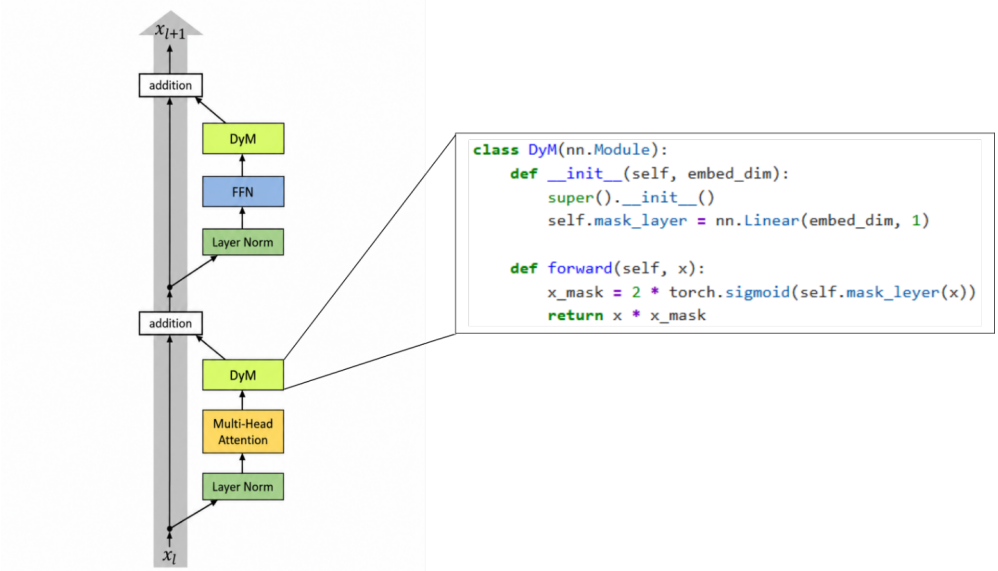

Figure 1: Integration of Dynamic Masking into a residual architecture. A functional layer computes a candidate update; DyM assigns an input-dependent coefficient before residual addition. In the Transformer implementation, independent DyM modules follow attention and feed-forward branches.

3.2 Architecture-dependent parameterization

The core DyM operation is agnostic to the network used to predict the coefficient. In ViT-B/16 and the antibody–antigen model we use the direct scalar form in Eq. (3), so one coefficient is produced for each content unit and broadcast across its hidden representation. In Pythia, attention- and MLP-side updates are modulated independently and the gate is hidden-dimension-wise. For a token update $\Delta\mathbf{h}_i \in \mathbb{R}^d$,

$$\mathbf{m}_i = 2\sigma(W_2\,\mathrm{GELU}(W_1\Delta\mathbf{h}_i + \mathbf{b}_1) + \mathbf{b}_2)\,, \tag{4}$$

where W_1 maps $d \to d/4$ and W_2 maps $d/4 \to d$. The $d/4$ bottleneck is an implementation choice rather than a defining part of DyM. It provides a low-dimensional nonlinear control representation: the candidate update is first compressed into a compact descriptor and then expanded into feature-specific coefficients. This limits gate capacity and parameter cost relative to a full-width feature-wise predictor while retaining nonlinear interactions among hidden features. We do not claim that $d/4$ is an optimal bottleneck width. The final projection W_2 and its bias are initialized to zero, so $\mathbf{m} = \mathbf{1}$ at initialization and the model begins from the unmodified residual computation.

The 3D segmentation experiment uses a third, scalar bottleneck parameterization. Each MONAI Swin UNETR block is wrapped after the block computation; its (B, D, H, W, C) output is flattened to spatial tokens and passed through LayerNorm, a $C \to \max(C/4, 16) \to 1$ MLP, GELU, and $2\sigma(\cdot)$. The resulting scalar is broadcast over channels and multiplies the spatial token. Thus the BraTS experiment tests spatially adaptive block-output modulation; unlike the Pythia and ViT branch-level implementations, it does not separately gate the internal attention and MLP residual branches.

The added parameter cost depends on gate granularity. Direct scalar gates add only 18,456 parameters to ViT-B/16 and 8,481 to the antibody–antigen model. The feature-wise Pythia bottleneck adds 1.58M parameters at 70M and 7.10M at 160M. In Swin UNETR, eight scalar bottleneck gates add 101,928 parameters, or 0.164% of the 62.19M-parameter backbone. Exact counts and formulas are reported in Appendix A.

3.3 Interpreting modulation across depth

The coefficient is first a computational quantity, not an attribution score. Relative to the unmodified residual operation, the magnitude of the change introduced by DyM can be written

$$E_i^{(l)} = |m_i^{(l)} - 1|\,\|\Delta\mathbf{h}_i^{(l)}\|_2. \tag{5}$$

Table 1: Pythia evaluation with and without DyM. HellaSwag and PIQA use normalized accuracy.

Model	HSwag	LAMB Acc.	LAMB PPL	PIQA	Wino
Orig-70M	.279	.238	105.459	.596	.521
DyM-70M	.268	.212	134.924	.587	.502
Orig-160M	.312	.363	29.040	.632	.503
DyM-160M	.300	.486	16.851	.616	.503

For compact visualization across K DyM-equipped locations, we summarize signed modulation in log space,

$$L_i = \frac{1}{K} \sum_{k=1}^{K} \log m_i^{(k)}, \tag{6}$$

and unsigned modulation magnitude as $D_i = K^{-1} \sum_k |\log m_i^{(k)}|$. These are descriptive summaries, not exact cumulative gains: residual updates are additive and later transformations depend on representations already modified by earlier layers.

4 Experiments

4.1 Experimental design

Across domains, we separate two questions: whether DyM changes predictive performance, and whether the learned control signal has interpretable computational structure. We therefore report standard task metrics first, then analyze the coefficients using either external labels or interventions. Structural biology and medical segmentation provide residue- or voxel-level labels that can be compared directly with learned modulation. Language and ImageNet do not provide a unique ground-truth importance map, so the primary evidence comes from perturbing inputs or the gate itself and measuring the resulting change in model behavior. This separation is important because a useful control signal need not improve every benchmark, and a coefficient that correlates with task structure need not be a conventional importance score. Accordingly, we treat predictive change, association with external structure, and causal sensitivity to intervention as distinct forms of evidence.

The recorded quantity depends on the gate parameterization. In Pythia-160M, each of the 12 Transformer layers contains separate attention- and MLP-side vector gates, giving 24 modulation locations. For token-level visualization, we average $\log m$ over hidden dimensions within each location and then across locations; signed averages preserve attenuation versus amplification, whereas $|\log m|$ measures strength of departure from the identity update. In the antibody–antigen model, scalar DyM is summarized over 33 locations for 216 complexes. In ViT-B/16, scalar DyM follows attention and MLP branches in all 12 blocks, again yielding 24 locations. In Swin UNETR, eight wrapped Swin blocks produce one scalar coefficient per spatial token; the stage dimensions are 48, 96, 192, and 384 with two blocks per stage. Additional implementation details and secondary analyses are given in the appendix.

4.2 Language modeling

We evaluated Pythia at 70M and 160M scales (Biderman et al., 2023) on HellaSwag, LAMBADA, PIQA, and WinoGrande. Effects were task and scale dependent (Table 1). At 70M, DyM modestly decreased performance. At 160M, however, LAMBADA accuracy increased from 0.363 to 0.486 and perplexity fell from 29.040 to 16.851, while HellaSwag and PIQA decreased slightly and WinoGrande was unchanged. Thus the paper does not rely on a universal performance-improvement claim; the stronger question is whether the learned modulation is structured and functionally used.

Linguistic structure and direct neutralization. After within-prompt centering and adjustment for token position, verbs, nouns, and adjectives showed more positive modulation than determiners and punctuation; they also occupied higher within-prompt modulation percentiles, and the separation was strongest on the MLP side (Table 2). To test whether DyM simply reproduces conventional token influence, we independently deleted context tokens and measured the change in fixed-target

Table 2: Position-adjusted token modulation by part of speech in Pythia-160M. Positive values indicate relatively stronger update incorporation after within-prompt centering.

POS	n	Signed DyM	Percentile	Attn.	MLP
VERB	43	.074	.759	.020	.155
NOUN	76	.054	.723	.035	.127
ADJ	15	.047	.692	.018	.125
DET	55	-.072	.156	-.071	-.213
PUNCT	42	-.087	.306	.024	-.108

Table 3: Effect of selectively neutralizing DyM in Pythia-160M.

Neutralized tokens	Mean ΔNLL	Median ΔNLL
Top 10% DyM magnitude	4.291	4.101
Position-matched random 10%	2.645	2.296
Bottom 10% DyM magnitude	1.380	0.587

log probability. Across 100 sequences, signed DyM had only a small position-adjusted association with signed deletion influence (Spearman $\rho = 0.059$, 95% CI $[0.035, 0.083]$), while modulation magnitude had essentially no positive association with absolute deletion influence ($\rho = -0.026$, 95% CI $[-0.052, -0.002]$). Position itself affected both quantities in opposite directions: absolute deletion influence increased with token position ($\rho = 0.301$), whereas DyM magnitude decreased ($\rho = -0.351$). These weak and differently structured correlations argue against treating the gate as a conventional deletion-based importance score.

We then intervened on the modulation itself. For 200 held-out sequences, we neutralized DyM ($m \leftarrow 1$) on the top 10% of context tokens ranked by modulation magnitude and compared them with the bottom 10% and position-matched random tokens. Neutralizing the top group increased fixed-target NLL by 4.291 on average, compared with 2.645 for position-matched random tokens and 1.380 for the bottom group (Table 3). The top-versus-position-matched difference was 1.647 (95% CI $[1.352, 1.956]$); top neutralization caused greater degradation in 79% of sequences ($p < 10^{-4}$, paired sign-flip test), and the top group exceeded the bottom group in 85% of sequences ($p < 10^{-4}$). Strongly attenuated tokens were especially consequential: neutralizing them increased target NLL by 4.200 versus 1.529 for strongly amplified tokens; for whole-sequence NLL, the corresponding increases were 3.784 and 0.647. Thus a small coefficient does not imply computational unimportance. Instead, strong attenuation can itself be a learned decision on which the downstream computation depends.

Gate ablation. To test whether the trained model merely benefits from generic residual scaling, we performed inference-time ablations on the Pythia-160M DyM checkpoint while keeping all backbone weights fixed. Full DyM retains the hidden-dimension-wise gate $m_{i,h}^{(l,b)}$, where i indexes tokens, h hidden dimensions, l layers, and $b \in \{\mathrm{attn}, \mathrm{MLP}\}$ the residual branch. The *token-scalar* condition removes hidden-dimension-specific control by replacing the vector with

$$\tilde{m}_i^{(l,b)} = \frac{1}{d} \sum_{h=1}^{d} m_{i,h}^{(l,b)}, \tag{7}$$

and broadcasting this value over h. The *branch-scalar* condition removes content dependence as well, replacing each module by $\bar{m}^{(l,b)} = \mathbb{E}_{x,i,h}[m_{i,h}^{(l,b)}]$, estimated from 50 held-out Pile validation batches. *Identity* sets every coefficient to one after training. Finally, *shuffled* DyM preserves every learned gate vector but randomly permutes its token assignment within the sequence; we average three independent shuffles. These are functional ablations of one trained checkpoint, not separately optimized architectures, so they isolate dependencies acquired by the trained DyM model rather than comparing independently trained gate families.

The reduced-gate conditions reveal a clear hierarchy. Collapsing the vector gate to a dynamic scalar per token increased LAMBADA perplexity from 16.851 to 73.249, showing that token dependence preserves part but not all of the learned computation. Replacing content-dependent gating with one fixed scalar per residual branch caused a much larger degradation (PPL 1066.118). Shuffling the full

Table 4: Inference-time gate ablations on trained Pythia-160M. Shuffled values are mean ± SD over three permutations.

Gate	LAMB PPL ↓	LAMB Acc.	HSwag	PIQA	Wino
Full DyM	16.851	.486	.300	.616	.503
Token scalar	73.249	.333	.274	.592	.499
Branch scalar	1066.118	.130	.265	.564	.474
Identity ($m = 1$)	1.35×10^7	.000	.265	.511	.506
Shuffled DyM	2.59×10^5	$.0058 \pm .0014$	$.259 \pm .002$	$.513 \pm .018$	$.507 \pm .006$

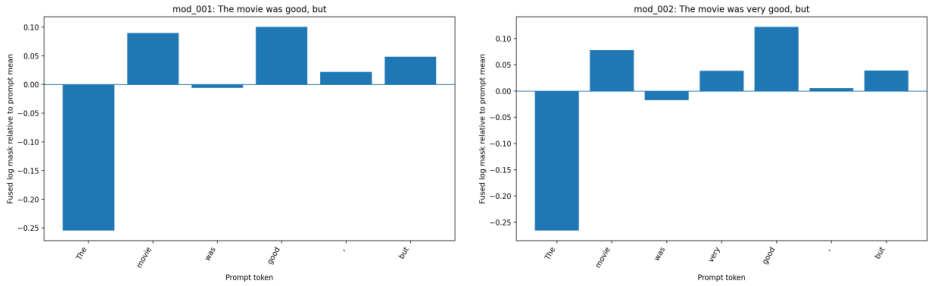

Figure 2: Context-dependent DyM modulation in Pythia for the closely related prompts "The movie was good, but" and "The movie was very good, but." Bars show fused log-DyM relative to the within-prompt mean. Shared downstream tokens change their modulation after the contextual edit, whereas the autoregressive prefix provides a causal-direction control.

gate vectors was more disruptive still despite preserving the gate vectors themselves. HellaSwag and PIQA showed the same broad ordering; WinoGrande remained near chance and did not distinguish the conditions. These results indicate that the trained model relies on average residual scaling, content-dependent token assignment, and hidden-dimension-specific modulation. The extreme identity failure is interpreted as post-training neutralization of a network that adapted to DyM during training, not as evidence that an independently trained conventional residual model is unstable.

Context sensitivity and depth. We additionally tested whether the modulation changes where altered linguistic context can actually affect an autoregressive representation. In controlled minimal pairs, identical tokens preceding a contextual modification showed almost no change in signed DyM (mean absolute difference 3.36×10^{-4}), whereas identical tokens downstream of the modification changed by 0.00591 on average, approximately 17.6× more. Modulation magnitude showed the same direction, with downstream changes about 12.7× larger than the prefix control. Because this analysis currently contains only three controlled pairs, we treat it as a directional sanity check rather than a standalone statistical result. Figure 2 illustrates one pair: adding "very" changes the modulation assigned to shared downstream tokens while leaving the causal prefix essentially unchanged. Calibration statistics also reveal a strong depth trend. Table 5 shows representative branch means from the 50-batch calibration set: early layers are strongly attenuating, coefficients approach one near layer 10, and the final layer is amplifying on average. This pattern suggests that DyM does not learn a uniform global shrinkage factor; instead, the operating regime changes substantially with depth. The functional meaning of that depth profile still requires targeted layerwise intervention.

4.3 Antibody–Antigen Interface Prediction

We next evaluated scalar DyM in an antibody–antigen interface model following prior work (Liu et al., 2026). Adding DyM modestly improved paratope AUPR from 0.743 to 0.751 while AUC remained 0.982; epitope AUC increased from 0.809 to 0.812 and AUPR from 0.458 to 0.472 (Table 6). More importantly for the present study, experimentally defined interface residues provide an external reference for the modulation signal.

Across 216 test complexes from the Paragraph dataset (Chinery et al., 2023), the analysis contained 119,236 non-interface and 8,201 interface residues. For each residue, we aggregate the scalar gate in log space across the 33 DyM-equipped locations, then compare the resulting distributions between

Table 5: Representative mean Pythia-160M gate coefficients by depth, estimated from 50 held-out Pile validation batches.

Layer	Attention DyM	MLP DyM
0	.165	.218
3	.344	.275
5	.446	.387
7	.596	.541
9	.810	.796
10	.974	1.024
11	1.162	1.291

Table 6: Antibody–antigen interface prediction with and without DyM.

Model	Para. AUC	Para. AUPR	Epi. AUC	Epi. AUPR
Original	.982	.743	.809	.458
DyM	.982	.751	.812	.472
Change	.000	+.008	+.003	+.014

experimentally defined interface and non-interface groups. Interface residues showed substantially lower fused log-DyM scores: -1.486 versus -1.108, a difference of -0.379. Exponentiating the mean log scores gives geometric-mean coefficients of 0.226 and 0.330, respectively (Table 7), so the typical multiplicative update coefficient is about 31.5% lower at interface residues. Because coefficients below one attenuate newly computed updates, interface residues undergo stronger attenuation on average. This does not imply lower importance; rather, interface status is associated with a distinct regime of representation control.

Figure 3 gives a spatial view. Prediction probability and last-layer attention are elevated near the interface, while fused DyM shifts in the opposite direction toward stronger attenuation. Attention characterizes information exchange among residue representations; DyM characterizes how strongly the resulting updates are incorporated. Their different spatial patterns therefore need not agree.

4.4 Computer vision

For ViT-B/16 on ImageNet-1K (Dosovitskiy et al., 2020; Deng et al., 2009), scalar DyM increased top-1 accuracy from 81.68% to 82.03%. The original and DyM models followed closely aligned training-loss trajectories (Appendix), suggesting that the added control did not qualitatively change the global optimization trajectory. We collected coefficients from the 24 attention- and MLP-side modulation locations and tested whether signed modulation was associated with the causal contribution of image patches. On 5,000 validation images, we rank patches using the clean forward pass, then replace the selected 5–30% of patches by the ImageNet mean. The predicted class c from the clean image is held fixed throughout the perturbation, and intervention strength is measured as $\Delta z_c = z_c(x) - z_c(x_{\setminus S})$, where S is the perturbed patch set. This avoids conflating a change in the identity of the top predicted class with a change in support for the original decision.

Unsigned magnitude did not act as a simple patch-importance score. For the top-D_i patches, mean logit reductions at 5%, 10%, 20%, and 30% intervention were .086, .208, .472, and .793, compared with .133, .273, .457, and .627 for random patches. Thus the highest-magnitude patches were less damaging than random at small intervention fractions and only exceeded random for broader perturbations. Signed modulation, however, showed a clear direction. Perturbing the most strongly amplified patches consistently produced the largest logit decrease (Table 8). At 30%, prediction-flip rates were 29.6% for strongly amplified patches, 11.8% for random patches, and 6.7% for strongly attenuated patches. The weaker effect of high-D_i patches at small intervention fractions is consistent with visual evidence being distributed across neighboring patches: removing a small number of individually extreme patches may leave redundant object evidence intact, whereas broader interventions disrupt the spatially distributed representation. This directional result complements language modeling, where strong attenuation was especially consequential, and reinforces that the meaning of high versus low coefficients is task dependent. A representative patch-level DyM map and last-layer attention visualization are provided in Appendix D; their different spatial patterns further illustrate that update incorporation and information exchange are distinct quantities.

Table 7: Fused DyM modulation for interface and non-interface residues.

Group	Residues	Mean log-DyM	Geom. mean
Non-interface	119,236	-1.108	.330
Interface	8,201	-1.486	.226
Difference	–	-0.379	–

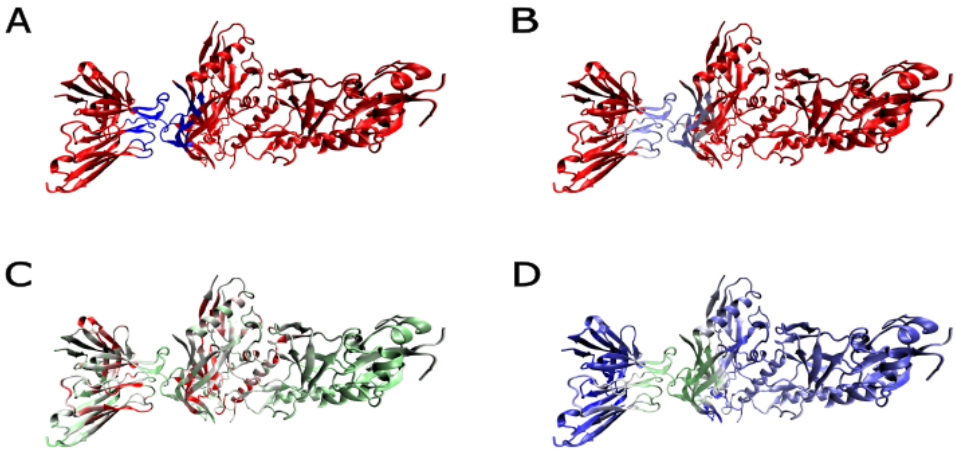

Figure 3: DyM modulation in antibody–antigen interface prediction. (A) Ground-truth interface; (B) predicted interface probability; (C) fused log-DyM, with green indicating stronger attenuation and purple stronger update incorporation; (D) last-layer attention. Interface residues tend toward stronger attenuation even where prediction and attention are elevated, illustrating that the signals describe different aspects of computation.

4.5 3D MEDICAL IMAGE SEGMENTATION

We further evaluated DyM with Swin UNETR (Hatamizadeh et al., 2021) for BraTS21 brain-tumor segmentation, following the MONAI Swin UNETR BraTS21 pipeline. The instantiated backbone uses four MRI input channels, three output channels, feature size 48, and the default four-stage Swin hierarchy with two blocks per stage. We wrap all eight Swin blocks with scalar DyM. Because the channel widths double across stages, the gate dimensions are 48, 96, 192, and 384. At each block, the 3D output is flattened into spatial tokens and the learned scalar gate is applied before restoring the volume layout. This implementation adds 101,928 trainable parameters, increasing the model from 62.192M to 62.294M parameters (0.164%).

Performance is reported with Dice overlap for tumor core (TC), whole tumor (WT), and enhancing tumor (ET). Mean Dice changed from 0.9040 to 0.9053: tumor core decreased from 0.9054 to 0.9005, whole tumor increased from 0.9263 to 0.9349, and enhancing tumor remained nearly unchanged (0.8802 to 0.8804). The learned modulation was nevertheless spatially structured. Average DyM coefficients were 1.1165 in tumor core, 1.0837 in whole tumor, and 1.1201 in enhancing tumor, compared with 0.8861 in non-tumor tissue, corresponding to approximately 1.22–1.26$\times$ enrichment. Thus, unlike the antibody–antigen setting, tumor regions tend to receive stronger modulation. Because the voxel-level tumor labels are defined independently of the learned gate, this comparison provides a second externally grounded spatial test. The direction is nevertheless opposite: tumor regions are relatively amplified, whereas antibody–antigen interface residues are relatively attenuated. A representative 3D visualization is given in Appendix E.

Cross-domain experimental pattern. The four domains provide complementary tests of the same control variable at different representational scales. The biological tasks supply external labels independent of DyM: antibody–antigen interface residues are more strongly attenuated than non-interface residues, whereas BraTS tumor regions are amplified relative to non-tumor tissue. Language

Table 8: Mean reduction in clean predicted-class logit after ImageNet patch intervention.

Fraction	Strong amplification	Random	Strong attenuation
5%	.247	.133	.073
10%	.535	.273	.157
20%	1.015	.457	.264
30%	1.498	.627	.389

Table 9: BraTS21 segmentation with and without DyM.

Region	Original Dice	DyM Dice	Mean DyM
Tumor core	.9054	.9005	1.1165
Whole tumor	.9263	.9349	1.0837
Enhancing tumor	.8802	.8804	1.1201
Non-tumor	–	–	.8861

and ImageNet instead use interventions. In Pythia, neutralizing the most strongly modulated tokens produces substantially larger NLL increases than neutralizing weakly modulated or position-matched random tokens, and the gate-ablation hierarchy shows that token alignment and hidden-dimension-specific structure are functionally used. In ImageNet, replacing strongly amplified patches produces larger clean-class logit reductions than either random or strongly attenuated patches. The common empirical pattern is therefore not a universal direction of modulation, but a non-uniform allocation of residual-update strength that tracks task-relevant computation under independent labels or causal perturbations.

5 Discussion

Across domains, DyM behaves as an update-control signal rather than a universal importance score. Interface residues and consequential language tokens can be strongly attenuated, whereas tumor regions and causally important image patches can be amplified. The Pythia ablations further show that the trained computation depends on content-aligned, hidden-dimension-specific modulation, not only on average branch scaling. Attention and DyM are therefore complementary: attention determines how information is exchanged to construct an update, while DyM regulates how strongly that update modifies the representation. Because residual updates are additive, fused scores remain descriptive summaries rather than exact cumulative gains. Two scope limitations are worth noting: the language experiments use Pythia models up to 160M parameters, and the gate parameterization is adapted to each architecture, so absolute coefficient magnitudes should not be compared directly across domains. The Pythia gate ablations are post-training interventions that identify structure used by the trained model; controlled retraining of alternative gate forms and larger models is a natural next step.

6 Conclusion

Dynamic Masking adds an explicit learned decision about how strongly newly computed information should modify each residual representation. Across language, proteins, images, and 3D medical volumes, the modulation is structured and computationally consequential, yet does not reduce to a monotonic importance score. The Pythia ablations show that useful modulation depends on content alignment and, for the vector-valued language implementation, hidden-dimension-specific control. We therefore view DyM as a general mechanism for adaptive representation refinement and as an inspectable control variable that can be tested directly through intervention.

AI USE STATEMENT

We used generative AI as a supporting tool during this project. It helped with code drafting and debugging, organizing literature, language editing, LaTeX formatting, and discussion of possible analyses. The scientific questions, experimental priorities, and final experiment choices were made by the authors. The structural-biology studies and language-model tests were designed by the authors; the 3D medical-image segmentation experiment and the DyM gate-ablation study were proposed during author discussions. All model training, evaluation, perturbation, ablation, and quantitative analyses were carried out by an author, and the reported numerical results were checked against the experimental outputs. AI-suggested code was reviewed, modified when needed, and tested before use. The manuscript was developed and revised by the authors with AI-assisted drafting and language editing. The authors made the final scientific interpretations and take responsibility for the paper.

REPRODUCIBILITY STATEMENT

The method equations and core intervention definitions are specified in the main text. Additional implementation details, parameter counts, language analyses, and domain-specific results are provided in the appendices. Code and configuration files used for the experiments will be included as anonymous supplementary material for review.

REFERENCES

Ashish Vaswani et al. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

Alexey Dosovitskiy et al. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint, 2020.

Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. Highway networks. arXiv preprint arXiv:1505.00387, 2015.

Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual reasoning with a general conditioning layer. *AAAI*, 2018.

Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, 2018.

Michael S. Ryoo et al. TokenLearner: Adaptive space-time tokenization for videos. *NeurIPS*, 2021.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. *ICML*, 2017.

Ramprasaath R. Selvaraju et al. Grad-CAM: Visual explanations from deep networks via gradient-based localization. *ICCV*, 2017.

Julius Adebayo et al. Sanity checks for saliency maps. *NeurIPS*, 2018.

Sarthak Jain and Byron C. Wallace. Attention is not explanation. *NAACL-HLT*, 2019.

Sarah Wiegreffe and Yuval Pinter. Attention is not not explanation. *EMNLP-IJCNLP*, 2019.

Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *NeurIPS*, 2017.

Jay DeYoung et al. ERASER: A benchmark to evaluate rationalized NLP models. *ACL*, 2020.

Sofia Serrano and Noah A. Smith. Is attention interpretable? *ACL*, 2019.

Jesse Vig et al. Bertology meets biology: Interpreting attention in protein language models. arXiv preprint arXiv:2006.15222, 2020.

Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1:206–215, 2019.

Tao Lei, Regina Barzilay, and Tommi Jaakkola. Rationalizing neural predictions. *EMNLP*, 2016.

Jasmijn Bastings, Wilker Aziz, and Ivan Titov. Interpretable neural predictions with differentiable binary variables. *ACL*, 2019.

Stella Biderman et al. Pythia: A suite for analyzing large language models across training and scaling. *ICML*, 2023.

Xing Liu, Jordan Kantorow, Amit K. Chattopadhyay, and Srirupa Chakraborty. Interpretable antibody–antigen structural interface prediction via adaptive graph learning and cyclic transfer. *bioRxiv*, 2026.

Lucy Chinery, Nancy Wahome, Iain Moal, and Charlotte M. Deane. Paragraph—antibody paratope prediction using graph neural networks with minimal feature vectors. *Bioinformatics*, 39, 2023.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. *CVPR*, 2009.

Ali Hatamizadeh, Vishwesh Nath, Yucheng Tang, Dong Yang, Holger R. Roth, and Daguang Xu. Swin UNETR: Swin Transformers for semantic segmentation of brain tumors in MRI images. In *MICCAI BrainLes Workshop*, 2021.

A Implementation Details and Parameter Overhead

Parameter counts were computed from the implementations used for each experiment. For the direct scalar gate used in ViT-B/16 and the antibody–antigen model, each DyM location contains a single $d \to 1$ affine projection and therefore adds $d + 1$ parameters. ViT-B/16 uses $d = 768$ at 24 attention/MLP locations, giving $24 \times 769 = 18{,}456$ added parameters (86,567,656 $\to$ 86,586,112). The antibody–antigen model uses $d = 256$ at 33 locations, giving $33 \times 257 = 8{,}481$ added parameters (18,764,034 $\to$ 18,772,515).

For Pythia, let the bottleneck width be $r = d/4$. A vector-valued gate contains $W_1 \in \mathbb{R}^{r \times d}$, $\mathbf{b}_1 \in \mathbb{R}^r$, $W_2 \in \mathbb{R}^{d \times r}$, and $\mathbf{b}_2 \in \mathbb{R}^d$, for $2dr + r + d$ parameters per location. This gives 1,580,544 added parameters for Pythia-70M ($d = 512$, 12 gate locations) and 7,100,928 for Pythia-160M ($d = 768$, 24 locations). The bottleneck is used as a capacity-controlled nonlinear parameterization of the feature-wise gate; it is not required by the general DyM formulation and we do not claim that the $1/4$ ratio is optimal.

The BraTS21 implementation follows the public `dym_segmentation` code: every Swin block in the MONAI Swin UNETR encoder is wrapped with a scalar bottleneck gate. A gate at channel width d contains LayerNorm ($2d$ parameters), $d \to h$ and $h \to 1$ affine layers, where $h = \max(d/4, 16)$, for $dh + 2d + 2h + 1$ parameters. With two blocks at each of $d \in \{48, 96, 192, 384\}$, the eight gates add

$$2 \sum_{d \in \{48,96,192,384\}} [d\,h(d) + 2d + 2h(d) + 1] = 101{,}928 \tag{8}$$

parameters. The corresponding Swin UNETR counts are 62,191,941 without DyM and 62,293,869 with DyM, an overhead of 0.164%. Unlike the branch-level Transformer placement used elsewhere, this wrapper multiplies the output of the complete Swin block after its internal residual operations; we therefore interpret the segmentation experiment as a block-level spatial-modulation test.

Table 10: Trainable parameter overhead introduced by DyM. Counts include all trainable gate parameters.

Model	Without DyM	With DyM	Increase
Pythia-70M	70.427M	72.007M	2.244%
Pythia-160M	162.323M	169.424M	4.375%
ViT-B/16	86.568M	86.586M	0.021%
Antibody–antigen	18.764M	18.773M	0.045%
Swin UNETR	62.192M	62.294M	0.164%

For the Pythia gate-ablation calibration, one scalar was estimated for every attention- and MLP-side DyM module from 50 held-out Pile validation batches. Mean coefficients increased markedly with depth: early branches were strongly attenuating (layer-0 attention 0.165; MLP 0.218), values approached the identity near layer 10 (0.974 and 1.024), and the final layer was on average amplifying (1.162 and 1.291). This depth profile is descriptive and is not used as a standalone importance measure.

B Additional Language Analyses

B.1 Part-of-speech organization

The main paper reports the representative position-adjusted part-of-speech statistics in Table 2. The MLP-side signal produces the clearest separation between content-bearing categories (verbs, nouns, adjectives) and function/punctuation categories in this analysis.

B.2 Context sensitivity

The main paper reports the controlled minimal-pair analysis and a representative visualization. The current set contains three pairs and is intended as a causal-direction sanity check rather than a population-level linguistic result.

C Additional Structural-Biology Results

With DyM, paratope AUC/AUPR changed from 0.982/0.743 to 0.982/0.751, while epitope AUC/AUPR changed from 0.809/0.458 to 0.812/0.472. These performance changes are modest relative to the primary mechanistic observation that experimentally defined interface residues occupy a distinct attenuation regime.

D Additional Vision Results

The original and DyM ViT-B/16 models showed closely aligned training-loss trajectories. Unsigned modulation magnitude was not a monotonic patch-importance score: at 5% and 10% intervention, perturbing the highest-magnitude patches produced less degradation than random perturbation, whereas the effect exceeded random at 30%.

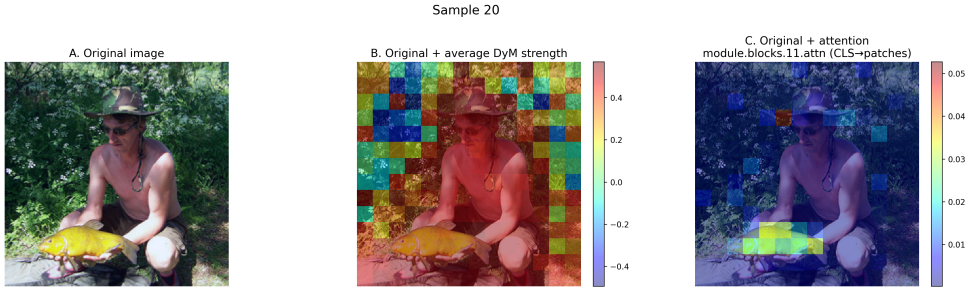

Figure 4: Spatial organization of DyM modulation and last-layer attention in ViT-B/16 for a representative ImageNet image.

E Additional 3D Segmentation Results

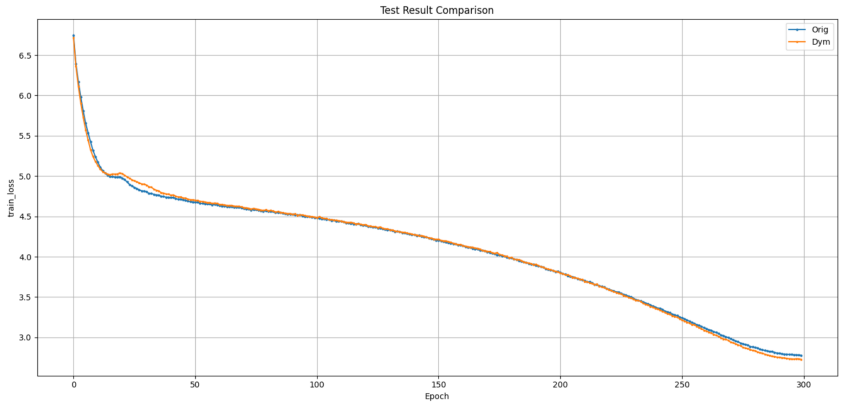

Figure 5: Training behavior of ViT-B/16 with and without DyM.

Model	TC	WT	ET	Mean
Original Swin UNETR	.9054	.9263	.8802	.9040
DyM Swin UNETR	.9005	.9349	.8804	.9053
Change	-.0049	+.0086	+.0002	+.0013

Region	DyM score	Enrichment vs. non-tumor
TC	1.1165	1.26×
WT	1.0837	1.22×
ET	1.1201	1.26×
Non-tumor	.8861	1.00×

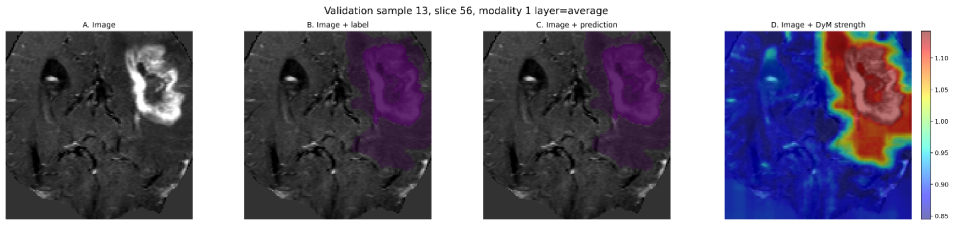

Figure 6: Spatial distribution of DyM in representative BraTS volumes. Tumor regions show systematically higher modulation than surrounding non-tumor tissue.