Constraint-Based Structural Attention in Graph Transformers

An Inverse-Optimisation-Based Approach

Cedric Pelsma

Master of Science Thesis

Delft Center for Systems and Control

Constraint-Based Structural Attention in Graph Transformers

An Inverse-Optimisation-Based Approach

Master of Science Thesis

For the degree of Master of Science in Systems and Control at Delft University of Technology

Cedric Pelsma

July 10, 2026

Faculty of Mechanical Engineering (ME) · Delft University of Technology

The work in this thesis was conducted in consultation with CogniChip

Delft University of Technology
Department of
Delft Center for Systems and Control (dcsc)

The undersigned hereby certify that they have read and recommend to the Faculty of Mechanical Engineering (ME) for acceptance a thesis entitled

Constraint-Based Structural Attention in Graph Transformers

by

Cedric Pelsma

in partial fulfillment of the requirements for the degree of

Master of Science Systems and Control

Dated: July 10, 2026

Supervisor(s):

Peyman Mohajerin Esfahani

Tolga Ok

Reader(s):

Amin Sharifi Kolarijani

Abstract

Many problems in science and engineering involve data that is naturally graph-structured: molecules, proteins, knowledge bases, road networks. Graph Transformers have become one of the leading model families for this kind of data. Because plain self-attention treats its input as an unordered set, the graph topology (edges, distances, connectivity patterns) has to be supplied to the model explicitly, either through structural encodings added to the node features or, more directly, through a learned bias added to the attention logits. In the latter case a structural term is added to the content score, where it competes with that score on equal footing and applies the same correction at every layer, regardless of whether content attention already respects the structural prior.

We take a different starting point. Softmax attention can itself be written as the solution to an entropy-regularised optimisation problem, which suggests treating structure not as another term in the objective but as a constraint on the attention itself. This thesis develops the idea into a graph transformer in which structural priors are imposed as per-pair inequality constraints on the attention distribution. Each correction is governed by a Lagrangian dual variable that the model carries across transformer layers, growing on pairs where content attention falls short of the structural prior and relaxing where the prior is already met.

The result is a graph transformer that learns structural constraints inside its attention mechanism rather than treating structure as a fixed additive term. Because the dual reacts to how far the realised attention falls from the structural target, the correction is active only where structure is genuinely under-served and silent elsewhere. The construction is agnostic to which pairwise graph signal plays the role of the constraint, and multiple signals can be combined without architecture changes. On the ZINC and LRGB Peptides molecular benchmarks, the adaptive constraint mechanism consistently improves over the matched baseline in which the dual is held fixed at unit scale.

Table of Contents

Acknowledgements

This work was partially supported by the Horizon Europe Pathfinder Open project RELIEVE-101099481 and by the European Research Council (ERC) project TRUST-949796.

I would like to thank a number of people whose help and support made this thesis possible.

My daily supervisor, Tolga Ok, deserves a special word of thanks for guiding me through this work. His guidance and the many discussions along the way taught me a great deal, not only about machine learning, but also about how to do research and how to manage a long-running project.

I am also grateful to my responsible supervisor, Peyman Mohajerin Esfahani, for his engagement throughout the project and for giving me the freedom to learn and to shape its direction, while providing insightful feedback.

A particular thanks goes to my brother Reindert for his help with a few software problems during the experimental stage of the thesis.

Finally, I would like to thank my family for their support throughout my studies, and in particular my girlfriend Lianne for her patience and encouragement during the writing of this thesis.

European Research Council
Established by the European Commission

Delft University of Technology
July 10, 2026

Cedric Pelsma

Chapter 1

Background

1-1 Introduction

A surprising fraction of the data that scientists and engineers care about is naturally graph-structured. Molecules are atoms connected by bonds, proteins fold according to residue contact patterns, knowledge bases are entities linked by relations, and road networks are intersections joined by roads [3, 36, 33]. Building models that learn well on such data has become one of the more active corners of machine learning research.

For most of the past decade the default tool has been the message-passing graph neural network (GNN). Each node aggregates information from its immediate neighbours at every layer, and stacking layers gradually widens the reach [3, 36]. This idea has been behind several notable successes: drug discovery [29], protein structure prediction [13], and biological network analysis [30]. Reach is also where it runs into trouble. Information moves one hop per layer, and stacking more layers tends to smooth node representations together rather than carrying more distant signal, so on larger graphs message-passing GNNs often miss the long-range dependencies that the task actually depends on [33].

Graph Transformers (GT) take a different route. They build on the Transformer architecture [32] that powers large language models like BERT [7] and GPT [5] (and the reinforcement-learning methods now used to sharpen their reasoning [21, 20]), where self-attention lets every token consult every other. In a graph transformer the same is true for nodes, and crucially the node features are handled separately from the graph structure: features flow through the ordinary attention pathway, while structural information enters through a dedicated channel that does not have to fight content for representational capacity. This split is what has put graph transformers on top of several benchmarks, especially the ones where long-range interactions matter and message-passing GNNs hit their receptive-field ceiling [34, 16, 11, 14, 28].

Constraints as a design principle

Self-attention can be read as the answer to an optimisation problem. A body of work shows that an attention layer is the optimiser of an implicit objective over the attention weights, and

works backwards from a given layer to recover that objective [12, 35, 23, 19, 25]. We use the same correspondence in the opposite direction. Rather than recover the optimisation problem behind an attention layer that already exists, we start from an optimisation problem of our own choosing and let its solution define the attention mechanism of a graph transformer.

This turns the optimisation problem into something we design: choosing what it optimises is choosing how attention behaves. Our choice is to state the behaviour we want as constraints. A constraint says what the attention distribution must satisfy and leaves the rest free, which is exactly the shape of a structural prior: it makes a demand on some node pairs and stays silent about the others.

Designing a model by prescribing its behaviour as constraints has a long track record in optimisation-based machine learning. Inverse optimisation [6, 26, 31, 24] works from observed solutions back to the objective or constraints that would produce them; we follow the forward direction, fixing the desired behaviour as constraints and reading the model's parameters off from them. Variants of this idea have been productive in control [1], routing [27], healthcare [2], and offline reinforcement learning [8]. The common thread is to build a prior into the architecture as a requirement on its output instead of hoping the model picks it up from data.

We apply this to the structural priors of a graph transformer. Each prior becomes a constraint that the attention distribution must respect, and the force with which it is enforced is set automatically, pair by pair, by how badly it is currently violated. What these structural priors are, and how graph transformers have supplied them so far, is what we turn to next.

Inductive biases for Graph Transformers

Where does this structural term come from, and why does attention need one at all? Self-attention scores the compatibility of two nodes purely from their content. It is permutation-invariant and treats the nodes as an unordered set, so on its own it cannot tell whether a pair of nodes is bonded, how many hops apart they are, or where they sit in the graph: the graph topology has to be supplied to the attention mechanism explicitly. A growing body of work on graph transformers does this by injecting structural inductive biases into the attention mechanism, and two broad strategies have taken hold.

The first adds structural positional encodings to the node features before attention is computed. Laplacian eigenvectors (LapPE) [9] and random-walk landing probabilities (RWSE) [10] both summarise each node's global position in the graph as a vector that is then appended to or mixed into its feature. This is flexible and architecture-agnostic, but once the encoding is folded into the node representation it can no longer say anything specific about a particular pair of nodes.

The second strategy adds a structural bias directly to the pre-softmax attention logit for each pair. Graphormer [34] uses shortest-path-distance embeddings together with edge-type encodings; Gradformer [15] applies an exponential decay over graph distance; spectral attention networks [14] turn Laplacian eigenvectors into pairwise biases; GPS [22] combines local message passing with global attention; and GRIT [16] introduces learnable RRWP-based pairwise encodings that evolve across layers. This is exactly the right place to inject pair-level structure, but every one of these biases carries the rigidity described above: the structural term

sits on the logit with a single global scale, shared across pairs and layers, with no way to ease off on the pairs where content attention already agrees with the prior.

This thesis builds on the structural-bias family. Graphormer-style additive biases have reached state-of-the-art performance on a range of molecular benchmarks [34], and they expose a single clean interface (an additive term on the attention logit) on which the adaptive per-pair control developed here can be built. Figure 1-1 places the two existing strategies next to the constraint-based approach of this thesis.

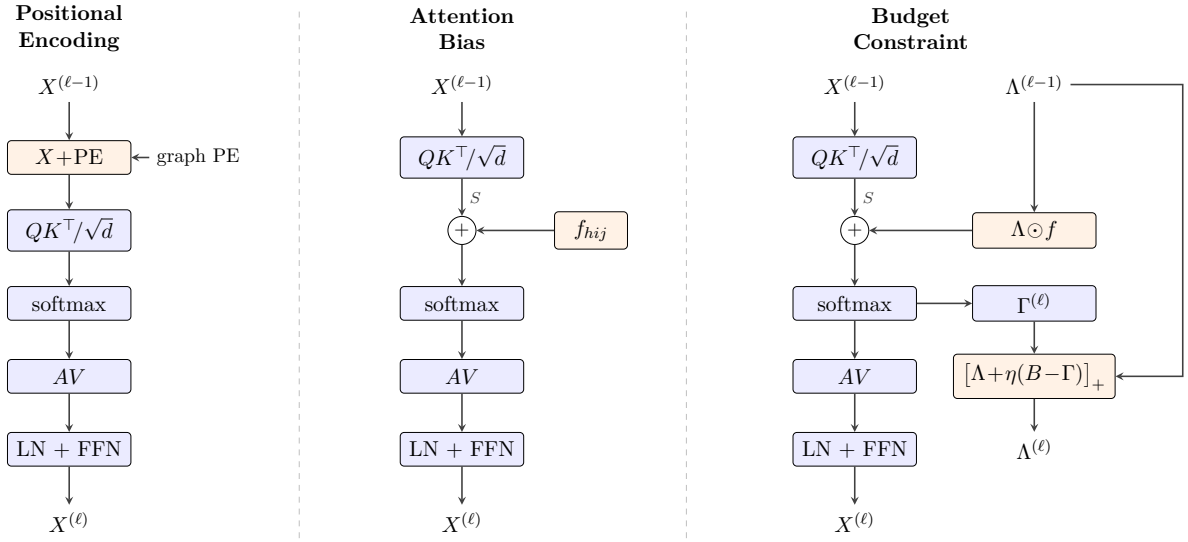


Figure 1-1: Three paradigms for injecting structural inductive bias into Transformer attention, each showing one full transformer layer. *Left*: structural positional encodings are added to the node features before any attention is computed. *Middle*: a fixed structural term f_{hij} is added directly to the pre-softmax attention logit. *Right* (this thesis): the structural signal $\Lambda \odot f$ is added to the logit with a per-pair strength $\Lambda^{(\ell)}$ that is updated after each layer. The dual path (orange) computes the head-averaged attention $\Gamma^{(\ell)}$ and applies the probability-budget step $[\Lambda+\eta(B-\Gamma)]_+$, so structural pressure grows where content attention falls below the budget and relaxes where it already exceeds it. The constraint-based construction sketched here is developed in full in Chapter 2.

Why graph priors fit the constraint framework

Pairwise structural priors on graphs (random-walk probabilities, shortest-path distances, bond types) are naturally phrased as statements about specific node pairs. "Atom i should attend to atom j at least as much as their random-walk proximity suggests" is already a per-pair requirement on the attention distribution; we just have to read it that way.

The split between content and structure also mirrors a familiar split in constrained optimisation. The objective is what the model is trying to do (content-based attention); the constraints are what it is obliged to respect (the structural prior). Wrapping the prior into a fixed additive bias collapses the two and forces content and structure to fight through a single global scalar. Keeping the prior as a feasibility constraint preserves the separation, and a per-pair dual variable turns each structural violation into a correction that adapts to the pair and to the layer it lives in.

The dual is zero for pairs where content and structure already agree, and grows only where a deficit persists. That is the kind of pair-level feedback a single global scale cannot give. As far as we are aware, no prior graph-transformer work treats its structural prior as a feasibility constraint whose dual is propagated across layers, carrying memory of past violations from one layer to the next.

Contributions

The thesis makes three contributions.

(*i*) **(An optimisation-based graph transformer.)** We design a graph transformer in which structural priors enter as feasibility constraints on the attention distribution rather than as additive biases. The construction is grounded in constrained optimisation: each structural prior comes with a dual variable, and the dual update rule that the model executes between layers can be read directly off the Lagrangian. Unrolling the solver that performs the forward pass recovers a stack of standard transformer layers, so the construction is not a departure from the usual architecture but a principled derivation of it.

(*ii*) **(A general interface for pairwise structural information.)** The mechanism does not care what the structural signal is. Any pairwise quantity can serve as the prior: shortest-path distances, bond types, random-walk probabilities, learned pairwise encodings. Several priors can be combined without changing the architecture, which makes the method a drop-in replacement for fixed structural biases in any graph transformer that already uses them.

(*iii*) **(Empirical evaluation on molecular-graph benchmarks.)** We evaluate the method on ZINC-12k and the LRGB Peptides benchmarks against matched fixed-bias baselines, and we identify and analyse a failure mode in which the constraint mechanism deactivates.

1-2 Related Work

This thesis is most directly related to two lines of work: Graphormer, which introduced structural biases for graph transformers, and GRIT, which extended them with cross-layer structural information. This section positions the proposed method relative to both.

1-2-1 Graphormer: Structural Biases Without Cross-Layer Feedback

Graphormer [34] demonstrated that standard Transformers can achieve strong performance on graph tasks when three structural encodings are added: a degree-based centrality encoding on node features, a spatial encoding that adds a learned scalar bias indexed by shortest-path distance to every attention logit, and an edge encoding that aggregates edge features along shortest paths.

The key architectural principle is to treat structural information as a *layer-shared additive logit bias*: for each node pair (i, j) and head h, the attention score is augmented by a structural term $b_\phi(i, j)$ whose values are learned from data but indexed only by the static graph topology, so the same learned bias is applied at every transformer layer. Graphormer achieves state-of-the-art results on molecular benchmarks including ZINC [34], confirming that encoding graph structure at the attention logit level is more effective than using only node-level positional encodings.

The limitation is that the structural bias is *non-adaptive at runtime*: although its values are learned, the same correction is applied at every layer, for every pair, regardless of whether the content attention already agrees with the structural prior. There is no feedback mechanism to increase structural pressure where the prior is being violated, nor to relax it where the prior is already satisfied.

1-2-2 GRIT: Cross-Layer Structural Information via Learned Encodings

GRIT [16] advances Graphormer by introducing a learnable pairwise encoding updated across layers. Instead of a fixed lookup table indexed by integer distances, GRIT uses a learned linear map from RRWP features to per-head attention biases, and applies a multi-layer perceptron (MLP) between transformer layers to update the pairwise structural representation based on the current node features. This allows the structural encoding to evolve with the network depth, giving the model richer cross-layer structural information.

GRIT therefore sits between layer-shared attention biases and full message passing: it has cross-layer pairwise information flow (unlike Graphormer) but the structural update is driven by learned feature transformations (unlike constraint attention, which drives the update from constraint violations).

1-2-3 This Work: Constraint-Driven Cross-Layer Dual Feedback

The present thesis occupies a different point in the design space. Like Graphormer, it uses a pairwise structural prior added to the attention logit. Like GRIT, it propagates structural information across transformer layers. But unlike both, the cross-layer update is not learned:

it is derived from the Lagrangian dual of a per-pair budget constraint, and it increases structural pressure precisely where the content attention violates the prior and relaxes it where the prior is satisfied.

Method	Structural bias	Cross-layer update	Update mechanism
Graphormer	learned bias, layer-shared	no	—
GRIT	learned pairwise encoding	yes	MLP on node features
This work	budget constraint	yes	Lagrangian dual (KL gradient)

Table 1-1: Positioning relative to Graphormer and GRIT. All three methods add pairwise structural information to the attention logit; they differ in whether and how that information is updated across layers.

The constraint-based update has an explicit optimisation interpretation that neither Graphormer nor GRIT provides: each layer performs one gradient-descent step on the KL divergence from the structural budget to the realised attention. The dual magnitude is interpretable at test time as a per-pair measure of how strongly the model had to enforce the structural prior.

> *Remark* 1-2.1 (Gap in the existing literature). Both Graphormer and GRIT treat structural information as an input to the attention computation. No prior work encodes the structural prior as an inequality constraint with per-pair Lagrangian dual variables that accumulate across layers based on constraint violations. The present thesis fills this gap.

1-3 Preliminaries

1-3-1 Notation

Scalars are written in plain type; vectors and matrices in bold. For a graph $G = (V, E)$ with $n = |V|$ nodes, the set of integers $\{1, \ldots, n\}$ is denoted $[n]$. The ℓ^2 norm of a vector x is $\|x\|_2$. For two distributions P, Q over a finite set, $\mathrm{KL}(P\|Q) = \sum_j P_j \log(P_j/Q_j)$ denotes the KL divergence with P as the reference distribution. The softmax over index j is $\mathrm{softmax}_j(z) = e^{z_j}/\sum_{j'} e^{z_{j'}}$. The probability simplex over n elements is $\Delta_n = \{p \in \mathbb{R}^n : p_j \geq 0,\ \sum_j p_j = 1\}$. The projection onto the non-negative reals is $[x]_+ := \max(x, 0)$. The number of attention heads is H, the key/query dimension is d_k, and the number of transformer layers is L. Structural signals are indexed by $k = 1, \ldots, K$. Node representations at layer ℓ are written $x_i^{(\ell)} \in \mathbb{R}^d$.

1-3-2 Graph-Structured Data and Structural Features

Let $G = (V, E)$ denote a graph with node feature vectors $x_i \in \mathbb{R}^{d_x}$ and optional edge features x_{ij}^e. Graph learning tasks require encoding structural properties alongside node content. The following definitions collect the structural encodings used in this thesis; the full pipelines that turn a raw graph into these pairwise signals are detailed in Appendix B.

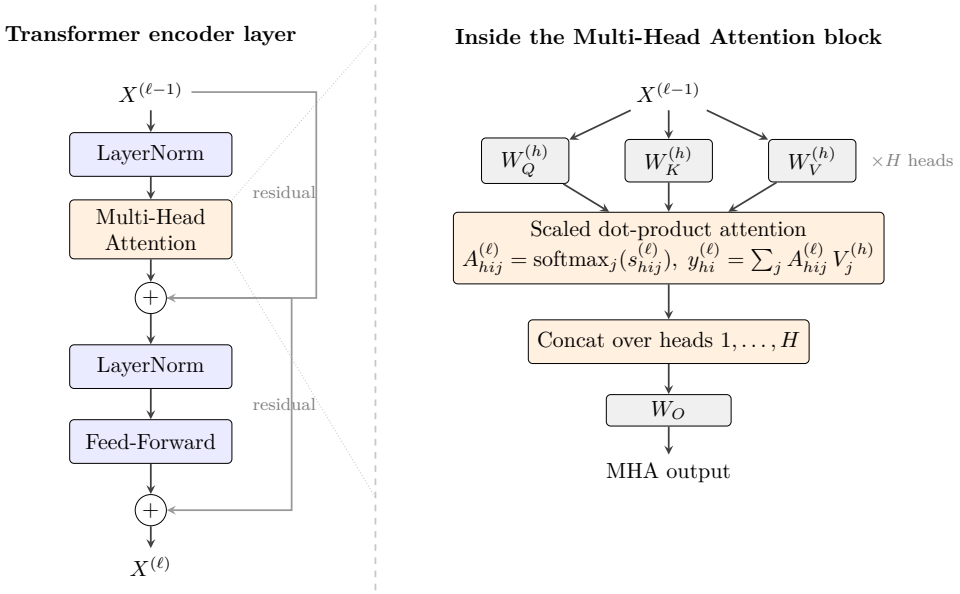


Figure 1-2: The transformer encoder layer (left) and the internals of multi-head attention (right). Inside the block, the input is projected into per-head queries $W_Q^{(h)}$, keys $W_K^{(h)}$, and values $W_V^{(h)}$ ($h = 1, \ldots, H$); each head computes scaled dot-product attention, and the H outputs are concatenated and passed through a final projection W_O.

Definition 1-3.1 (Relative Random-Walk Probability (RRWP))**.** Given the random-walk transition matrix $\mathbf{P} \in \mathbb{R}^{n \times n}$ of G, detailed in Appendix B-3, the RRWP feature between nodes i and j is the length-T vector of k-step landing probabilities $\mathrm{RRWP}_{ij} = ([\mathbf{P}^0]_{ij}, [\mathbf{P}^1]_{ij}, \ldots, [\mathbf{P}^{T-1}]_{ij}) \in \mathbb{R}^T$, where $\mathbf{P}^0 = \mathbf{I}$. Each power $\mathbf{P}^k$ is row-stochastic, with

$\sum_j [\mathbf{P}^k]_{ij} = 1$ for every i, so the random-walk signal yields a per-pair distribution over keys that can serve as a structural budget.

Definition 1-3.2 (Shortest-Path Distance Bias (SPD))**.** The shortest-path distance $\mathrm{SPD}(i, j)$ is the length of the shortest path between nodes i and j in G. Graphormer [34] encodes this as a learned scalar embedding added to the attention logit, effectively shifting the score of every pair by a function of their graph distance.

Other standard encodings include Laplacian positional encodings [9] and edge/path descriptors aggregated along shortest paths [34].

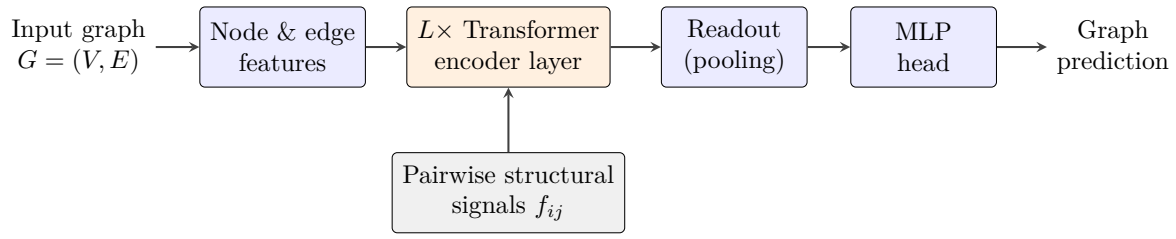


Figure 1-3: The complete graph-transformer pipeline, read left to right. A graph G is mapped to node and edge features; pairwise structural signals f_{ij} (shortest-path distance, bond type, RRWP) are fed into the attention of L transformer encoder layers (Figure 1-2); a permutation-invariant readout pools the nodes into a graph embedding, and an MLP head produces the prediction.

1-3-3 Transformer and Multi-Head Attention

The transformer layer [32] consists of a multi-head self-attention module followed by a feed-forward network. For node representations $x_i^{(\ell)} \in \mathbb{R}^d$ at layer ℓ, each attention head h computes content scores, attention weights, and an aggregated output:

Definition 1-3.3 (Content Attention Score)**.** For head h and layer ℓ, the content score between node i (query) and node j (key) is

$$s_{hij}^{(\ell)} = \frac{(W_Q^{(h)} x_i^{(\ell)})^\top (W_K^{(h)} x_j^{(\ell)})}{\sqrt{d_k}}, \tag{1-1}$$

where $W_Q^{(h)}, W_K^{(h)} \in \mathbb{R}^{d_k \times d}$ are learned projections. Attention weights and value aggregation are then $A_{hij}^{(\ell)} = \mathrm{softmax}_j(s_{hij}^{(\ell)}), \quad y_{hi}^{(\ell)} = \sum_j A_{hij}^{(\ell)} W_V^{(h)} x_j^{(\ell)}$.

Figure 1-2 shows the full encoder layer—multi-head attention and a feed-forward network, each wrapped in layer normalisation and a residual connection—alongside the internals of the multi-head attention block.

Graph structure can enter this computation as additive pairwise biases on $s_{hij}^{(\ell)}$, as masking, or—as developed in Chapter 2—as constraints on the realised attention distribution.

1-3-4 The Full Model

Figure 1-3 places the attention computation above in the context of the complete pipeline used throughout this thesis. The pairwise structural signals that feed the encoder are constructed

as described in Appendix B, and the dataset and training configuration for each benchmark is given in Appendix A.

Chapter 2

Constraint Attention: Structural Budget Constraints

Figure 1-1 in the introduction contrasted two established strategies for injecting structural information into graph transformers: positional encodings that modify node features before attention, and additive biases that shift attention logits with a fixed structural term. Both treat the structural prior as input to an otherwise unconstrained optimisation, competing with content through a globally tuned scale. This chapter develops a third approach: encoding structural priors as *constraints* on the attention distribution and deriving their scale automatically from Lagrangian duality.

2-1 The Structural Optimisation Problem

Standard softmax attention for query row i and head h can be written as the solution of an entropy-regularised optimisation:

$$\max_{\{A_{hij}\geq 0,\, \sum_j A_{hij}=1\}} \sum_j A_{hij}\, s^{(\ell)}_{hij} + \mathcal{H}(A_{hi\cdot}), \tag{2-1}$$

where $s^{(\ell)}_{hij}$ is the content score (Section 1-3-3), $\mathcal{H}$ is the Shannon entropy, and the unique solution is the softmax distribution. The following lemma states this identity formally; it is the building block used repeatedly in the derivations of this chapter.

Lemma 2-1.1 (Entropy-regularised softmax [19, 17])**.** *For any vector $z \in \mathbb{R}^n$, the entropy-regularised problem*

$$\max_{p\in\Delta_n} \Phi(p,z), \qquad \Phi(p,z) := \sum_j p_j\, z_j + \mathcal{H}(p), \quad \mathcal{H}(p) = -\sum_j p_j \log p_j,$$

admits a unique optimum $p^* \in \Delta_n$ given by $p^*_j = \mathrm{softmax}_j(z) = e^{z_j}/\sum_{j'} e^{z_{j'}}$, and the

optimal value is the log-partition function:

$$\Phi(p^*, z) \;=\; \sum_j p_j^*\, z_j + \mathcal{H}(p^*) \;=\; \log \sum_j e^{z_j}.$$

Proof. Form the Lagrangian for the simplex constraint $\sum_j p_j = 1$ with multiplier μ: $\mathcal{L}(p, \mu) = \sum_j p_j z_j + \mathcal{H}(p) + \mu(1 - \sum_j p_j)$. Stationarity $\partial\mathcal{L}/\partial p_j = z_j - \log p_j - 1 - \mu = 0$ gives $p_j^* = e^{-(1+\mu)}\, e^{z_j}$, which is strictly positive (so the inequality $p_j \geq 0$ is inactive). Imposing $\sum_j p_j^* = 1$ fixes $e^{-(1+\mu)} = (\sum_{j'} e^{z_{j'}})^{-1}$, hence $p_j^* = \mathrm{softmax}_j(z)$. Substituting $\log p_j^* = z_j - \log \sum_{j'} e^{z_{j'}}$ into $\Phi(p^*, z) = \sum_j p_j^*(z_j - \log p_j^*)$ and using $\sum_j p_j^* = 1$ gives $\Phi(p^*, z) = \log \sum_j e^{z_j}$. Uniqueness follows from strict concavity of the Shannon entropy on Δ_n [19, 17]. □

Lemma 2-1.1 provides the entry point for adding structural knowledge. The objective captures what the model is trying to achieve (content-aligned attention), while structural requirements enter as *constraints* on what the attention distribution is required to respect. Encoding the prior as a constraint keeps the roles of content and structure cleanly separated, exactly the principle outlined in the introduction.

The content score $s_{hij}^{(\ell)}$ is the inner product between query and key projections of the node features (Section 1-3-3); it measures how strongly the learned features at nodes i and j align in the attention computation. *Structural* information is different in nature: it captures how strongly two nodes are compatible based on the graph topology alone, independently of the learned features. Concrete examples include shortest-path distance (Definition 1-3.2) and random-walk landing probabilities (Definition 1-3.1). Throughout this chapter the structural prior is encoded as a per-pair scalar $f_{hij}^{(k)}$ derived from such graph-based information, and the constraint will require the realised attention to respect it.

2-1-1 Structural Budgets

The framework places no restriction on the choice of structural signal: any per-pair scalar derived from graph topology can serve as $f_{hij}^{(k)}$, as long as it can be converted into a row-stochastic distribution over keys. Constraint attention is therefore agnostic to whether the signal is hand-crafted (shortest-path distance, bond type), spectral (Laplacian eigenvectors), or random-walk-based (RRWP probabilities); the same dual machinery applies unchanged. The only design choice for a given signal is the mapping that turns it into a budget.

Definition 2-1.2 (Structural Budget)**.** Let $f_{hij}^{(k)} \in \mathbb{R}$ be the logit-space structural score for signal k, head h, and node pair (i, j). The per-head budget, the head-averaged budget, and the score-weighted target are

$$b_{hij}^{(k)} = \mathrm{softmax}_j(f_{hij}^{(k)}) = \frac{\exp(f_{hij}^{(k)})}{\sum_{j'} \exp(f_{hij'}^{(k)})}, \tag{2-2}$$

$$B_{ij}^{(k)} = \frac{1}{H} \sum_{h=1}^{H} b_{hij}^{(k)}, \tag{2-3}$$

$$\hat{B}_{ij}^{(k)} = \sum_{h=1}^{H} b_{hij}^{(k)}\, f_{hij}^{(k)}. \tag{2-4}$$

Normalising the structural signal through softmax to obtain $b_{hij}^{(k)}$ is the design choice that makes the rest of the chapter work: it places the structural target in the same probability space as attention, and Section 2-2 will show that it leads to a dual update with the same shape as a Graphormer-style additive bias, which is both interpretable and numerically stable.

Each signal in this thesis is delivered to the constraint as a per-head, per-pair structural score $f_{hij}^{(k)}$ (a logit), from which the budget $b_{hij}^{(k)} = \mathrm{softmax}_j(f_{hij}^{(k)})$ follows by (2-2). The signal families differ only in how the score $f_{hij}^{(k)}$ is constructed:

Example 1: Graphormer-style signals. The SPD embedding $f_{hij}^{(\mathrm{SPD})} = e_{\mathrm{SPD}(i,j)}$ and the bond-type encoding $f_{hij}^{(\mathrm{bond})} = W_{\mathrm{bond}}\, e_{\mathrm{cat}(i,j)}$ are produced directly by a learned per-distance or per-category embedding (Appendices B-1 and B-2).

Example 2: RRWP signals. The T-step random-walk landing probabilities are log-transformed and passed through a learned linear projection that produces the score $f_{hij}^{(\mathrm{RRWP})}$ (Appendix B-3).

Full pipelines that convert raw graph topology into the structural score $f_{hij}^{(k)}$ for each signal family used in this thesis are described in Section 2-5-1 and Appendix B. Constraint attention also supports combining multiple structural signals: each signal k carries its own dual variable $\Lambda_{ij}^{(k)}$ and contributes its own correction term to the attention logit, as detailed in Section 2-3.

2-1-2 The Constrained Primal

With budgets defined, constraint attention augments (2-1) with a per-pair structural constraint on the realised attention, parameterised by the score-weighted target $\hat{B}_{ij}$ from Definition 2-1.2:

$$\boxed{\begin{aligned} &\max_{\{A_{hij}\}} \ \textstyle\sum_{h,i,j} A_{hij}\, s_{hij}^{(\ell)} + \mathcal{H}(A) \\ &\text{s.t.} \quad \textstyle\sum_h A_{hij}\, f_{hij} \ \geq \ \hat{B}_{ij} \quad \text{(structural constraint)} \\ &\qquad \textstyle\sum_j A_{hij} = 1,\ A_{hij} \geq 0 \quad \text{(simplex)} \end{aligned}} \tag{2-5}$$

Remark 2-1.3 (Why the structural constraint is not row-wise). Unlike the simplex constraint, which normalises each query row separately, the structural constraint is stated per pair (i, j) and couples the heads through the sum $\sum_h$. A structural prior is a statement about a specific pair, so it carries one dual variable per pair and aggregates that pair's attention across all heads rather than per row. The head sum is deliberately left unnormalised: no $1/H$ appears on either side of $\sum_h A_{hij} f_{hij} \geq \hat{B}_{ij}$ because the factor cancels. The head-averaging $1/H$ re-emerges only later, through the step-size rescaling of Section 2-3, which is what turns the constraint into the head-averaged form $\Gamma_{ij} \geq B_{ij}$.

For later use we name the two aggregations of the realised attention that the constraint and the dual update act on.

Definition 2-1.4 (Head-averaged and score-weighted attention)**.** For attention weights A_{hij}, head count H, and structural score f_{hij}, the *head-averaged attention* and the *score-*

weighted aggregation are

$$\Gamma_{ij} \;=\; \frac{1}{H}\sum_{h=1}^{H} A_{hij}, \qquad \tilde{\Gamma}_{ij} \;=\; \sum_{h=1}^{H} f_{hij}\, A_{hij}. \tag{2-6}$$

Γ_{ij} is row-stochastic and lives in the same probability space as the head-averaged budget B_{ij}; $\tilde{\Gamma}_{ij}$ is the f-weighted quantity that appears on the left-hand side of the structural constraint.

The structural constraint has f appearing on both sides: it weights the realised attention on the LHS and, through $\hat{B}_{ij} = \sum_h b_{hij} f_{hij}$, it weights the budget on the RHS. This bilinear-in-f shape is what gives the algorithm its additive structure: in Section 2-2 we derive that the inner KKT solution has the form $A^*_{hij} = \mathrm{softmax}(s_{hij} + \Lambda_{ij} f_{hij})$, the same shape as a Graphormer-style fixed additive bias [34] (which adds f to the logit at unit scale), with the per-pair dual Λ_{ij} replacing that fixed unit weight by a quantity that adapts per pair and per layer. Solving the primal drives the realised attention to match the structural budget B_{ij} entrywise: content scores and the dual-weighted structural term combine through the softmax to produce attention that respects the budget while staying close to the content-only distribution, as illustrated for a representative graph in Figure 2-1.

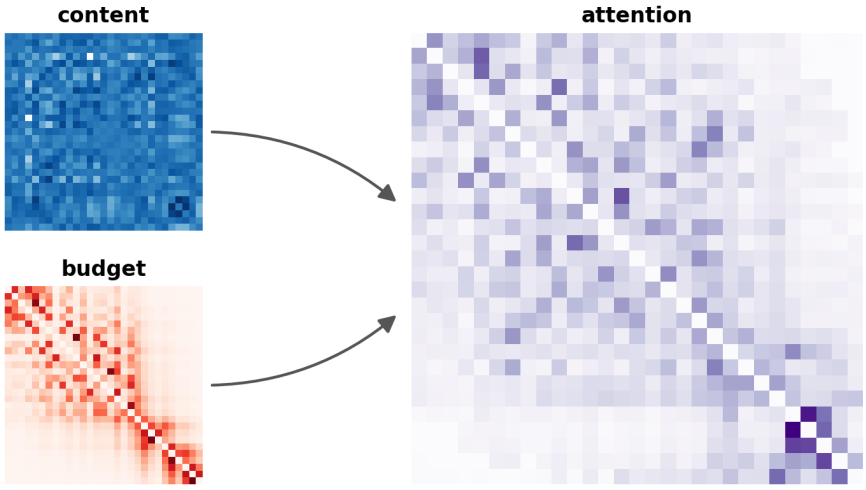


Figure 2-1: Content scores s_{hij} and the dual-weighted structural term $\Lambda^{(\ell)}_{ij} f_{hij}$ combine through the softmax to produce realised attention that respects the structural budget while minimising distortion from content-only attention.

The forward pass and dual update derived in Section 2-2 solve (2-5) in a single iteration each, and the transformer applies them layer by layer. Each transformer layer is therefore one iterative step on this optimisation problem: the forward pass computes the inner KKT solution at the current Λ, and the dual variable is updated between layers based on how far the realised attention falls from the budget. Figure 2-2 shows the resulting layer-by-layer evolution of attention for a representative query node, with keys sorted by their budget value B_{qj}: as the dual grows on persistently under-attended pairs, attention mass concentrates onto the high-budget keys, while keys already above budget see their dual relax and content attention dominates. The geometric counterpart, namely the trajectory of the per-layer attention in the probability simplex relative to the constraint hyperplane $\Gamma = B$, is given in Figure 2-5

and analysed in Section 2-4.

The hyperplane $\Gamma = B$ itself is not yet visible from the primal (2-5), whose constraint is the score-weighted inequality $\sum_h A_{hij} f_{hij} \geq \hat{B}_{ij}$. We will see in Section 2-3 that the factor f appearing on both sides of this inequality can be cancelled by an appropriate scaling of the dual step size; the constraint then reduces to the head-averaged inequality $\Gamma_{ij} \geq B_{ij}$, which row-stochasticity of Γ and B turns into the equality $\Gamma = B$ shown in Figure 2-5.

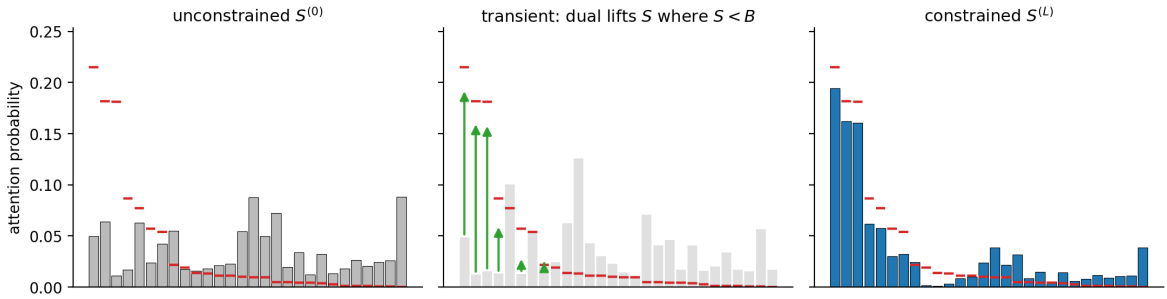


Figure 2-2: Attention from query node q across layers, keys sorted by budget B_{qj}. Mass concentrates on high-budget keys as Λ_{qj} grows; for keys already above budget the dual decreases and content attention dominates.

2-2 Forward Pass and Dual Update

This section derives the closed-form forward pass and the dual update rule directly from (2-5). The structure of the algorithm is straightforward: the inner KKT condition gives the attention update at fixed Λ, and projected gradient descent on the dual gives the update rule for Λ between layers.

For each pair (i, j) introduce a non-negative dual variable $\Lambda_{ij} \geq 0$ (one per structural signal k; superscript k dropped for clarity). The Lagrangian relaxation of (2-5), with additional multipliers μ_{hi} for the simplex constraints, is

$$\mathcal{L}(A, \Lambda, \mu) = \sum_{h,i,j} A_{hij}\, s^{(\ell)}_{hij} - A_{hij} \log A_{hij} + \sum_{i,j} \Lambda_{ij} \left(\sum_h A_{hij}\, f_{hij} - \hat{B}_{ij} \right) + \sum_{h,i} \mu_{hi} \left(1 - \sum_j A_{hij} \right). \tag{2-7}$$

2-2-1 Forward Pass

For fixed Λ and μ, the Lagrangian (2-7) separates over the per-row attention vectors $A_{hi\cdot} \in \Delta_n$: collecting the terms that involve A_{hij} gives, for each (h, i),

$$\max_{A_{hi\cdot} \in \Delta_n} \sum_j A_{hij}\, (s^{(\ell)}_{hij} + \Lambda_{ij}\, f_{hij}) + \mathcal{H}(A_{hi\cdot}),$$

after dropping the constant μ_{hi} (which does not affect the argmax over the simplex). This is exactly the entropy-regularised optimisation of Lemma 2-1.1 with augmented logits

$$z^{(\ell)}_{hij} \;=\; s^{(\ell)}_{hij} + \Lambda^{(\ell)}_{ij}\, f_{hij}. \tag{2-8}$$

The lemma yields the unique closed-form inner solution

$$A^{*(\ell)}_{hij} \;=\; \mathrm{softmax}_j\big(z^{(\ell)}_{hij}\big). \tag{2-9}$$

The dual Λ_{ij} acts as a per-pair scale on the structural score f_{hij}, producing the augmented attention logit that the forward pass actually computes.

2-2-2 Dual Function and Update

The dual function evaluates the Lagrangian at the inner optimum: $g(\Lambda) = \max_{A\in\Delta} \mathcal{L}(A, \Lambda, \mu^*(\Lambda))$. Its closed form follows in two steps. First, the inner maximum over A of each per-row term in (2-7) is given by Lemma 2-1.1 applied at the augmented logit $z_{hij} = s_{hij} + \Lambda_{ij} f_{hij}$:

$$\max_{A_{hi\cdot}\in\Delta_n} \;\sum_j A_{hij}\, z_{hij} + \mathcal{H}(A_{hi\cdot}) \;=\; \log \sum_j e^{z_{hij}}.$$

Second, the remaining Λ-dependent term in (2-7) is $-\sum_{i,j} \Lambda_{ij}\hat{B}_{ij}$, where $\hat{B}_{ij}$ is the *score-weighted target* defined in (2-4). Unlike B_{ij} (a row-stochastic average of per-head budgets), $\hat{B}_{ij}$ is the score-weighted aggregation $\sum_h b_{hij} f_{hij}$: the value the constraint LHS would take if the attention matched the budget distribution exactly. Summing the two contributions over all (h, i) gives the proposition.

Proposition 2-2.1 (Dual function and gradient)**.** *The dual function has the explicit closed form*

$$g(\Lambda) \;=\; \sum_{h,i} \log \sum_j \exp\big(s_{hij} + \Lambda_{ij}\, f_{hij}\big) \;-\; \sum_{i,j} \Lambda_{ij}\, \hat{B}_{ij}. \tag{2-10}$$

It is convex in Λ: each log-sum-exp term is a convex function of its argument $s_{hij} + \Lambda_{ij} f_{hij}$, which is affine in Λ, and the remaining term $-\sum_{ij} \Lambda_{ij}\hat{B}_{ij}$ is linear, so the dual problem $\min_{\Lambda\geq 0} g(\Lambda)$ is a convex programme. Its gradient is

$$\frac{\partial g}{\partial \Lambda_{ij}} \;=\; \sum_h f_{hij}\, A^*_{hij} \;-\; \hat{B}_{ij}. \tag{2-11}$$

Proof. The closed form follows from the two-step derivation above, in which Lemma 2-1.1 provides the inner maximum and its log-partition value. Convexity follows because each $\log \sum_j \exp(\cdot)$ is convex in its argument $s_{hij} + \Lambda_{ij} f_{hij}$, which is affine in Λ, while the subtracted term $\sum_{ij} \Lambda_{ij}\hat{B}_{ij}$ is linear in Λ. For the gradient, differentiate the log-sum-exp $\log \sum_{j'} \exp(z_{hij'})$ with $z_{hij'} = s_{hij'} + \Lambda_{ij'} f_{hij'}$: the standard identity gives $\partial/\partial\Lambda_{ij} \log \sum_{j'} \exp(z_{hij'}) = f_{hij}\, A^*_{hij}$, where $A^*_{hij} = \mathrm{softmax}_j(z_{hij})$ by Lemma 2-1.1. Summing over heads yields the first term of (2-11); differentiating $-\Lambda_{ij}\hat{B}_{ij}$ gives $-\hat{B}_{ij}$. □

Projected gradient descent on the convex dual then yields the iteration

$$\Lambda^+_{ij} \;=\; \big[\Lambda_{ij} - \eta\, \partial g/\partial \Lambda_{ij}\big]_+ \;=\; \Big[\Lambda_{ij} + \eta\big(\hat{B}_{ij} - \textstyle\sum_h f_{hij}\, A^*_{hij}\big)\Big]_+, \tag{2-12}$$

with $[\cdot]_+$ enforcing $\Lambda \geq 0$. Together, (2-9) and (2-12) are the forward pass and dual update implied by the single primal (2-5). Strong duality means that minimising g recovers the exact

primal optimum; convergence guarantees follow from convexity of g once a suitable step size is chosen.

The gradient $\sum_h f_{hij} A^*_{hij} - \hat{B}_{ij}$ is the score-weighted deficit at the realised attention: positive when the f-weighted aggregation of A falls short of the target $\hat{B}$, negative when it exceeds it. The dual grows on under-served pairs and shrinks on over-served ones, projected back to the non-negative orthant when the constraint is slack.

Figure 2-3 summarises the single-layer dataflow implied by the forward pass (2-9) and the dual update (2-12). The content path augments the content logits with the structural term $\Lambda^{(\ell)} \odot f$, passes them through softmax, and feeds the result into value aggregation and the FFN. The dual path reads off the score-weighted aggregation $\tilde{\Gamma}^{(\ell)}_{ij}$ (Definition 2-1.4) after the softmax and applies $[\Lambda^{(\ell)} + \tilde{\eta}(\hat{B} - \tilde{\Gamma}^{(\ell)})]_+$ to produce the dual for the next layer. The step size $\tilde{\eta}$ is itself a function of f: this dependence is the subject of Section 2-3 and is what makes the dual update identifiable for a learnable f.

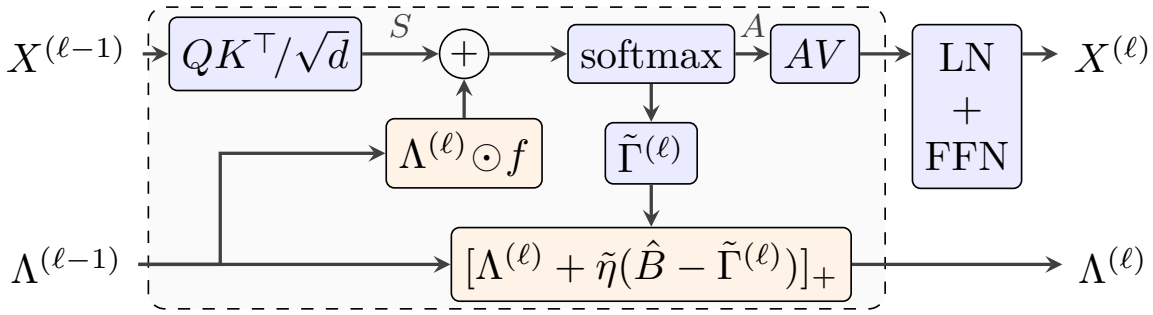


Figure 2-3: One transformer layer under constraint attention. The content path (shaded) augments logits with $\Lambda^{(\ell)} \odot f$ before softmax; the dual path (orange) reads the score-weighted aggregation $\tilde{\Gamma}^{(\ell)}_{ij} = \sum_h f_{hij} A^{(\ell)}_{hij}$ and applies $[\Lambda^{(\ell)} + \tilde{\eta}(\hat{B} - \tilde{\Gamma}^{(\ell)})]_+$ to update the dual for the next layer. The step size $\tilde{\eta}$ depends on f; under the head-uniform reduction of Section 2-3 this score-weighted update collapses to the head-averaged $[\Lambda + \eta(B - \Gamma)]_+$ of Figure 1-1.

2-3 Learning the Structural Correction

The forward pass and dual update derived in Section 2-2 constitute a complete algorithm at fixed f: on each forward pass, the entire model solves a well-defined constrained optimisation problem in which Λ adjusts to satisfy the structural constraint of (2-5). In practice, the structural signal f is learned together with the rest of the model from task supervision. This section explains why the raw dual update (2-12) interacts badly with learning f, and shows how a pair-specific step-size rescaling fixes the issue without changing the forward pass. The rescaled update is the form the algorithm actually uses, and it admits a clean KL-divergence interpretation that links back to the primal.

2-3-1 Identifiability and Step-Size Rescaling

Learning f from task supervision is an inverse-optimisation problem: the model observes the realised attention A^* and must infer the structural signal f that produced it. Classical

identifiability requires f to enter the Lagrangian in a role that the dual variable Λ cannot silently absorb.

Why the raw update fails identifiability. The attention the model produces, $A^*_{hij} = \mathrm{softmax}_j(s_{hij} + \Lambda_{ij} f_{hij})$, depends on the structural signal f and the dual Λ *only through their product* $\Lambda_{ij} f_{hij}$. Scaling

$$f \to c\, f, \qquad \Lambda \to \Lambda / c \qquad (c > 0) \tag{2-13}$$

leaves this product, and hence the realised attention A^*, exactly unchanged. Observing A^* therefore determines only the product Λf, never the scale of f on its own. The raw update (2-12) does nothing to break this scaling invariance: its target $\hat{B}(f) = \sum_h b_{hij} f_{hij}$ scales with f in step with the constraint left-hand side $\sum_h A_{hij} f_{hij}$, so the whole constraint is, to first order, invariant under the same rescaling (2-13). The magnitude of f is thus unrecoverable from observation alone.

The scale problem. The raw update has a second, practical defect: the size of its steps depends on the arbitrary scale of f. The step direction is the deficit $\sum_h f_{hij} A_{hij} - \hat{B}_{ij}$, and since both terms carry a factor of f, this deficit is measured in f-weighted units rather than in probability units. A large f then produces large dual steps (and a small equilibrium Λ), while a small f produces vanishingly small steps, even though the requirement we actually care about is a statement about the attention distribution, which does not change when f is rescaled to cf. What we want instead is a dual driven by a deficit measured directly in probability space: the gap $B_{ij} - \Gamma_{ij}$ between the target distribution B and the realised head-averaged attention Γ, which is independent of the scale of f. The rescaling introduced next delivers exactly this.

The remedy. A per-pair step size that itself depends on f restructures the update so that f appears only on the RHS. The following proposition is the central result of the chapter: a single, data-dependent rescaling makes the dual update identifiable in f while leaving the forward pass unchanged and keeping the algorithm within the single optimisation problem of (2-5).

Proposition 2-3.1 (Rescaled dual update)**.** *Define the pair-specific step size*

$$\tilde{\eta}_{ij} \;=\; \frac{\eta}{H\, f_{ij}}, \tag{2-14}$$

and assume the head-uniform regime $f_{hij} = f_{ij} > 0$, which keeps the step size $\tilde{\eta}_{ij}$ positive and lets the score-weighted constraint be divided through by $H f_{ij}$ without reversing its direction. Then projected gradient descent on the score-weighted dual g of Proposition 2-2.1, applied with step $\tilde{\eta}_{ij}$ on component Λ_{ij}, takes the closed form

$$\boxed{\Lambda^+_{ij} \;=\; \left[\Lambda_{ij} + \eta\,(B_{ij} - \Gamma_{ij})\right]_+}, \tag{2-15}$$

where Γ_{ij} is the head-averaged attention (Definition 2-1.4) and $B_{ij} = \frac{1}{H}\sum_h b_{hij}$ the head-averaged budget. In this form Λ depends on f only through the target B on the RHS; the LHS quantity Γ is f-independent, so f becomes identifiable from observation of the realised attention A^*.

Proof. From Proposition 2-2.1 the gradient of g is $\partial g/\partial \Lambda_{ij} = \sum_h f_{hij}\, A^*_{hij} - \hat{B}_{ij}$. Under head-uniform f the budget b_{hij} becomes head-independent ($b_{hij} = B_{ij}$) and the target factorises as $\hat{B}_{ij} = \sum_h b_{hij} f_{hij} = H f_{ij} B_{ij}$. The gradient then collapses to

$$\frac{\partial g}{\partial \Lambda_{ij}} \;=\; f_{ij} \sum_h A^*_{hij} \;-\; H f_{ij} B_{ij} \;=\; H f_{ij}\,(\Gamma_{ij} - B_{ij}).$$

The pair-specific gradient step is $\Lambda_{ij} - \tilde{\eta}_{ij}\, \partial g/\partial \Lambda_{ij} = \Lambda_{ij} - (\eta/H f_{ij}) \cdot H f_{ij}(\Gamma_{ij} - B_{ij}) = \Lambda_{ij} + \eta(B_{ij} - \Gamma_{ij})$. Projecting onto the non-negative orthant gives (2-15).

The identifiability claim follows from the form of the update. The realised attention $A^* = \mathrm{softmax}(s + \Lambda f)$ depends on f only through the product Λf, so the rescaling $f \to cf$, $\Lambda \to \Lambda/c$ of (2-13) leaves A^*, and hence the head-averaged Γ, unchanged. The target $B = \mathrm{softmax}(f)$, however, is not scale-invariant: $\mathrm{softmax}(cf) \neq \mathrm{softmax}(f)$ for $c \neq 1$. The rescaling therefore changes the right-hand side of the update while leaving the left-hand side Γ fixed, so the fixed point $\Gamma = B$ no longer holds and the two scalings of f become distinguishable—in contrast to the raw update (2-12), where f enters both sides through the product Λf and the rescaling cancels. □

The rescaling is the operating condition under which dual ascent on the primal of (2-5) is also a well-posed identifiability procedure for f.

In practice f_{hij} varies across heads. The rescaling (2-14) is then a notational equivalence rather than a runtime computation: the algorithm applies the rescaled update (2-15) directly, which is well-defined regardless of the sign or magnitude of f. The equivalence is exact in the head-uniform regime; outside it, Remark 2-3.6 below still guarantees fixed-point feasibility of the iteration.

The forward pass is unchanged. For multiple signals, each k contributes its own correction term:

$$A^{(\ell)}_{hij} \;=\; \mathrm{softmax}_j\left(s^{(\ell)}_{hij} + \sum_k \Lambda^{(k,\ell)}_{ij}\; f^{(k)}_{hij} \right), \tag{2-16}$$

and each dual $\Lambda^{(k)}$ runs its own rescaled update on its own budget. When all $\Lambda^{(k)} = 0$ the algorithm reduces to standard content-only attention; under-attended pairs accumulate dual mass and amplify their corresponding structural score.

Connection to Graphormer. The augmented attention logit $s_{hij} + \sum_k \Lambda^{(k)}_{ij} f^{(k)}_{hij}$ is structurally identical to the Graphormer attention [34], which simply adds the structural term $\sum_k f^{(k)}_{hij}$ to the content logit at unit scale. Constraint attention recovers Graphormer by setting $\Lambda^{(k)}_{ij} \equiv 1$ for all (i, j, ℓ) and turning off the dual update ($\eta = 0$): the rescaled update of Proposition 2-3.1 then leaves Λ frozen at unit value and the logit reverts to $s_{hij} + \sum_k f^{(k)}_{hij}$. Constraint attention is therefore the adaptive generalisation of Graphormer-style additive bias: the same forward pass, but with the unit bias weight replaced by a per-pair, per-layer dual that is set by the structural constraint of (2-5) rather than fixed by the architecture.

Figure 2-4 visualises the two pathways created by the rescaled update. On the left, f flows through the augmented logit and receives task-loss gradient with Λ held constant. On the

right, the same f defines the budget B, and the deficit $B-\Gamma$ analytically updates Λ via (2-15). Both sides share f but no gradient crosses between them, which is exactly the structural separation the identifiability argument requires.

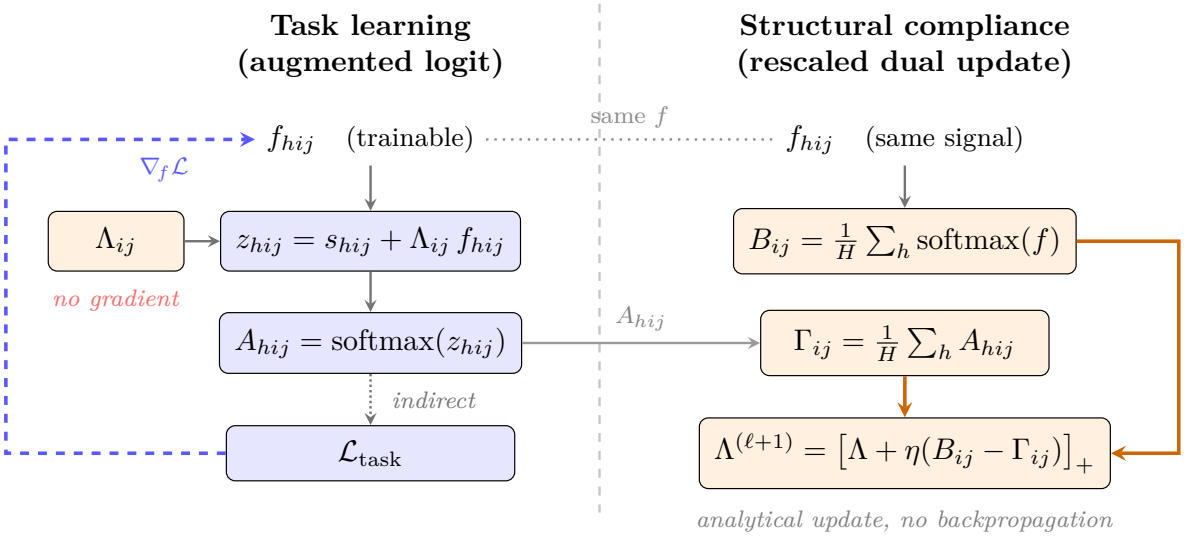


Figure 2-4: The two decoupled pathways implied by the rescaled dual update. *Left*: task loss trains f through the augmented logit; Λ enters the forward pass but carries no gradient. The dotted arrow from A_{hij} to $\mathcal{L}_{\text{task}}$ marks that the realised attention does not enter the loss directly but only through the downstream value aggregation, FFN, and prediction head of the transformer. *Right*: f defines budget B and the deficit $B - \Gamma$ updates Λ analytically via (2-15). Both sides share f but no gradient crosses the boundary, which is exactly what identifiability requires.

2-3-2 Connection to KL Divergence

The rescaled iteration (2-15) has a second interpretation that ties it directly back to the dual function g of Section 2-2. Under the rescaling, projected gradient descent on g coincides with gradient descent on the KL divergence between the budget and the realised attention. No second dual function or auxiliary Lagrangian is needed; the entire chapter still works with the single score-weighted dual of Proposition 2-2.1, and the rescaling is the operating condition under which the KL link is exact.

The connection is most transparent in the head-uniform regime $f_{hij} = f_{ij}$, which is the regime that makes the rescaling (2-14) an identity rather than an approximation. Throughout this subsection we assume this regime; the implementation runs the rescaled iteration (2-15) directly and inherits the fixed-point feasibility of Remark 2-3.6 below regardless.

Under the chosen rescaling $\tilde{\eta}_{ij} = \eta/(H f_{ij})$ of Proposition 2-3.1, the dual gradient direction in (2-11) factorises as $H f_{ij}(\Gamma_{ij} - B_{ij})$ in the head-uniform regime; the factor $H f_{ij}$ cancels exactly against the rescaling, so the effective deficit driving the iteration is the head-averaged $B_{ij} - \Gamma_{ij}$. Equivalently, the original score-weighted constraint $\sum_h A_{hij} f_{hij} \geq \hat{B}_{ij}$ of (2-5) is expressed in the rescaled coordinates as $\Gamma_{ij} \geq B_{ij}$ (dividing both sides by $H f_{ij} > 0$ using $\hat{B}_{ij} = H f_{ij} B_{ij}$). Row-stochasticity of both sides then forces it to be an equality.

Proposition 2-3.2 (Inequality is an equality)**.** *Let B_{ij} be row-stochastic ($\sum_j B_{ij} = 1$) and $\Gamma_{ij} = \frac{1}{H}\sum_h A_{hij}$ be the head-averaged attention, which is also row-stochastic. Then $\Gamma_{ij} \geq B_{ij}$ for all j if and only if $\Gamma_{ij} = B_{ij}$ for all j.*

Proof. ($\Leftarrow$) Trivial. ($\Rightarrow$) Summing $\Gamma_{ij} - B_{ij} \geq 0$ over j gives $\sum_j(\Gamma_{ij} - B_{ij}) = 0$. Since every term is non-negative and their sum is zero, each term must vanish. □

The feasibility set of (2-5) in the head-uniform regime $f_{ij} > 0$ is therefore the convex hyperplane $\{A : \Gamma(A) = B\}$. Over this set the primal selects the point closest to the unconstrained content-only attention in KL divergence.

Proposition 2-3.3 (Minimum-distortion interpretation)**.** *Assume the head-uniform regime $f_{hij} = f_{ij} > 0$. Let $A^{\text{free}}_{hij} = \text{softmax}_j(s_{hij})$ denote the unconstrained content-only attention. The primal* (2-5) *is equivalent to the minimum-KL projection of A^{free} onto the constraint set $\{A : \Gamma(A) = B\}$:*

$$\{A^*_{hij}\} \;=\; \underset{\substack{A\in\Delta\\ \Gamma(A)=B}}{\arg\min} \sum_{h,i} \text{KL}(A_{hi\cdot} \,\|\, A^{\text{free}}_{hi\cdot}). \tag{2-17}$$

Proof. By Lemma 2-1.1, $A^{\text{free}}_{hij} = \text{softmax}_j(s_{hij})$, so $\log A^{\text{free}}_{hij} = s_{hij} - \log Z_{hi}$ with $Z_{hi} = \sum_j \exp(s_{hij})$. Expanding the negative KL:

$$-\text{KL}(A_{hi\cdot}\|A^{\text{free}}_{hi\cdot}) = \sum_j A_{hij}(s_{hij} - \log Z_{hi}) + \mathcal{H}(A_{hi\cdot}) = \sum_j A_{hij}s_{hij} + \mathcal{H}(A_{hi\cdot}) - \log Z_{hi}.$$

Summing over h, i and dropping the constant $\sum_{h,i} \log Z_{hi}$, the entropy-regularised content objective of (2-5) equals $-\sum_{h,i} \text{KL}(A_{hi\cdot}\|A^{\text{free}}_{hi\cdot})$ up to a constant. Under head-uniform $f > 0$, the score-weighted constraint reduces to $\Gamma(A) = B$ by Proposition 2-3.2. Maximising the primal is therefore equivalent to minimising the total KL divergence over $\{A : \Gamma(A) = B\}$. Uniqueness follows from strict convexity of KL and convexity of the constraint set. □

The clean minimum-KL projection holds *only* in the head-uniform regime $f > 0$, because that is what lets the score-weighted constraint $\sum_h A_{hij} f_{hij} \geq \hat{B}_{ij}$ be divided through by $Hf_{ij} > 0$ and rewritten as $\Gamma \geq B$, which row-stochasticity promotes to the equality $\Gamma = B$ (Proposition 2-3.2). Outside this regime the constraint stays the score-weighted inequality $\{A : \sum_h A_{hij} f_{hij} \geq \hat{B}_{ij}\}$, which does not collapse to the hyperplane $\{\Gamma = B\}$, so projecting onto its feasible region is not the plain minimum-KL projection of (2-17). The entropy-regularised content objective projects in KL precisely onto the f-independent hyperplane $\{\Gamma = B\}$ that the head-uniform regime exposes.

The model satisfies the structural target with the *minimum possible distortion* to content attention (see Figure 2-1 in Section 2-1 for a representative visualisation). A fixed logit bias does not have this property: it applies the same correction at every layer regardless of whether the budget is already met, while constraint attention leaves such pairs unchanged ($\Lambda = 0$) and corrects only where a genuine deficit persists.

The min-distortion view also connects directly to the dual function g of Proposition 2-2.1: under the rescaling and head-uniform f, g coincides with the total KL divergence up to a constant in Λ.

Proposition 2-3.4 (Score-weighted dual equals total KL up to a constant)**.** *Let $g(\Lambda)$ be the score-weighted dual function of Proposition 2-2.1, and let*

$$F(\Lambda) \;=\; \sum_{h,i} \mathrm{KL}(B_{i\cdot} \,\|\, A^*_{hi\cdot}(\Lambda))$$

be the total KL divergence with B as reference and the realised attention $A^*_{hij} = \mathrm{softmax}_j(s_{hij} + \Lambda_{ij} f_{hij})$ as hypothesis. Assume the head-uniform regime $f_{hij} = f_{ij}$. Then

$$F(\Lambda) \;=\; g(\Lambda) \;+\; C, \tag{2-18}$$

where $C = H \sum_{i,j} B_{ij} \log B_{ij} - \sum_{h,i,j} B_{ij}\, s_{hij}$ depends only on B and s, not on Λ. Consequently $\nabla_\Lambda g = \nabla_\Lambda F$, and projected gradient descent on g under the rescaling (2-14) *is equivalent to the same preconditioned descent on the total KL objective.*

Proof. Under head-uniform $f_{hij} = f_{ij}$ the per-head budget $b_{hij} = \mathrm{softmax}_j(f_{ij})$ is head-independent, so $b_{hij} = b_{ij} = B_{ij}$ and $\hat{B}_{ij} = \sum_h b_{ij} f_{ij} = H f_{ij} B_{ij}$. By Lemma 2-1.1 applied at $z_{hij} = s_{hij} + \Lambda_{ij} f_{ij}$, $\log A^*_{hij} = s_{hij} + \Lambda_{ij} f_{ij} - Z_{hi}$ with $Z_{hi} = \log \sum_j \exp(s_{hij} + \Lambda_{ij} f_{ij})$ the log-partition. Expanding F and using $\sum_j B_{ij} = 1$ and the head-independence of B and f:

$$\begin{aligned} F(\Lambda) &= \sum_{h,i,j} B_{ij} \log B_{ij} - \sum_{h,i,j} B_{ij} (s_{hij} + \Lambda_{ij} f_{ij} - Z_{hi}) \\ &= \underbrace{H \sum_{i,j} B_{ij} \log B_{ij} - \sum_{h,i,j} B_{ij}\, s_{hij}}_{C} - H \sum_{i,j} B_{ij} \Lambda_{ij} f_{ij} + \sum_{h,i} Z_{hi}. \end{aligned}$$

From the definition of g in (2-10),

$$g(\Lambda) \;=\; \sum_{h,i} Z_{hi} \;-\; \sum_{i,j} \Lambda_{ij} \hat{B}_{ij} \;=\; \sum_{h,i} Z_{hi} \;-\; H \sum_{i,j} \Lambda_{ij} f_{ij} B_{ij},$$

so $F(\Lambda) - g(\Lambda) = C$, independent of Λ. □

Proposition 2-3.4 has an immediate algorithmic consequence: the rescaled update of Proposition 2-3.1 performs gradient descent on the KL divergence F at the same time as it descends the dual g.

Corollary 2-3.5 (Rescaled update descends total KL)**.** *Under the rescaling* (2-14) *and head-uniform f, the iteration* (2-15) *is preconditioned projected gradient descent on the total KL divergence F of Proposition 2-3.4:*

$$\Lambda^+_{ij} - \Lambda_{ij} \;=\; -\tilde{\eta}_{ij}\, \partial F / \partial \Lambda_{ij} \;=\; \eta\, (B_{ij} - \Gamma_{ij}),$$

projected onto $\Lambda \geq 0$. In particular, a single iteration decreases both g and F.

Proof. With $A^*_{hij} = \mathrm{softmax}_j(z_{hij})$ from Lemma 2-1.1 at $z_{hij} = s_{hij} + \Lambda_{ij} f_{hij}$, the log-softmax derivative $\partial \log A^*_{hij'} / \partial z_{hij} = \delta_{j'j} - A^*_{hij}$ yields $\partial F / \partial \Lambda_{ij} = \sum_h f_{hij} (A^*_{hij} - B_{ij})$. Under head-uniform f this equals $H f_{ij} (\Gamma_{ij} - B_{ij})$, which agrees with $\partial g / \partial \Lambda_{ij}$ in the same limit (so $\nabla g = \nabla F$, as already established in Proposition 2-3.4). Multiplying by the pair-specific step $\tilde{\eta}_{ij} = \eta / (H f_{ij})$ collapses $H f_{ij} (\Gamma_{ij} - B_{ij})$ to $\eta (\Gamma_{ij} - B_{ij})$, and projection onto the non-negative orthant gives (2-15). Both g and F decrease because the iteration is gradient descent on each (they differ only by a constant). □

Finally, it is worth recording what the rescaled iteration enforces once it settles, since the head-uniform caveats above appealed to it. The statement is the standard KKT characterisation of fixed points of projected gradient descent on the non-negative orthant [4, §5.5.2], specialised to our update; it does not require the head-uniform regime.

> *Remark* 2-3.6 (Fixed-point feasibility). Let $\Lambda^* \geq 0$ be a fixed point of the rescaled update $\Lambda^+ = [\Lambda + \eta(B - \Gamma(\Lambda))]_+$ with $\eta > 0$, where $\Gamma(\Lambda) = \frac{1}{H}\sum_h \mathrm{softmax}_j(s_{hij} + \Lambda_{ij} f_{hij})$ is the head-averaged attention of the score-weighted forward pass (2-9). Splitting the fixed-point equation by the sign of Λ^*_{ij} gives complementary slackness: where $\Lambda^*_{ij} > 0$ the projection is inactive and $\Gamma^*_{ij} = B_{ij}$; where $\Lambda^*_{ij} = 0$ it forces $\Gamma^*_{ij} \geq B_{ij}$. Both cases yield $\Gamma^*_{ij} \geq B_{ij}$, which row-stochasticity of Γ^* and B promotes to the equality $\Gamma^*_{ij} = B_{ij}$ entrywise (Proposition 2-3.2). Under head-uniform f, where $\hat{B}_{ij} = H f_{ij} B_{ij}$, this is exactly the original constraint of (2-5).

Two consequences carry into the rest of the thesis. First, wherever the rescaled iteration settles, the constraint is enforced exactly and the dual is non-zero only on the pairs where the constraint binds; this is the static guarantee that the head-uniform caveats above relied on, now stated in general. Second, the slack branch is precisely the condition under which the mechanism switches itself off: on pairs where content attention already meets the budget ($\Gamma^*_{ij} \geq B_{ij}$), the deficit $B_{ij} - \Gamma^*_{ij}$ is non-positive, so nothing drives the dual and it settles at $\Lambda^*_{ij} = 0$, leaving the structural correction inactive. This is the mechanism behind the deactivation failure mode analysed empirically in Section 3-0-4. Whether the iteration *reaches* a fixed point at all is a separate question, governed by the cross-layer dynamics analysed next.

2-4 Cross-Layer Behaviour

The derivations in the previous sections hold for a *fixed* content score matrix s_{hij}. In a transformer forward pass the content score $s^{(\ell)}_{hij}$ instead changes at every layer, for two reasons. First, the node features $x^{(\ell)}$ evolve from layer to layer. Second, the transformer is *not* weight-tied: each layer carries its own query, key, and value projections, so it would produce a different content score even on identical features. The algorithm therefore applies one rescaled dual step to a *sequence of distinct* optimisation problems, one per layer. This section analyses what the dual accumulates across these problems and what assumptions are required for that accumulation to be useful.

2-4-1 Cross-Layer Memory

The dual variable $\Lambda^{(k,\ell)}_{ij}$ is a scalar for each ordered pair (i, j) and signal k. Across an L-layer forward pass, the model maintains $O(KN^2)$ state, one value per ordered pair per signal. This is qualitatively different from standard node features, which provide $O(N)$ states: the dual stores which specific pairs have been persistently under-attended, not just which nodes have been underrepresented.

The dual state resets at the beginning of each forward pass and accumulates across layers. Under the rescaled update with $\Lambda^{(k,0)} = 0$, the state approximates an integrated structural

deficit:

$$\Lambda_{ij}^{(k,\ell)} \approx \Big[\eta \sum_{m=0}^{\ell-1} (B_{ij}^{(k)} - \Gamma_{ij}^{(m)})\Big]_+. \tag{2-19}$$

The dual grows for pairs that are persistently under-attended and shrinks for pairs where content attention already exceeds the budget. A fixed logit bias applies the same structural pressure at every layer; constraint attention modulates that pressure up or down based on how well the realised attention tracks the structural budget. This redistribution was previewed in Figure 2-2 (Section 2-1): mass concentrates onto high-budget keys as the corresponding dual grows, while keys already above budget see their dual relax.

> *Remark* 2-4.1 ($O(N^2)$ information flow). Each dual variable $\Lambda_{ij}^{(k,\ell)}$ records whether the specific pair (i, j) has under-attended its budget, which is $O(N^2)$ information that cannot be losslessly encoded in node features. Aggregating pairs into node features collapses the distinction between "pair (i, j) violated its budget" and "pair (i, k) violated its budget", which are different structural events. The dual matrix keeps both explicit, at the cost of $O(KN^2)$ additional scalars per forward pass.

2-4-2 Layerwise Local Optimality and the Cross-Layer Assumption

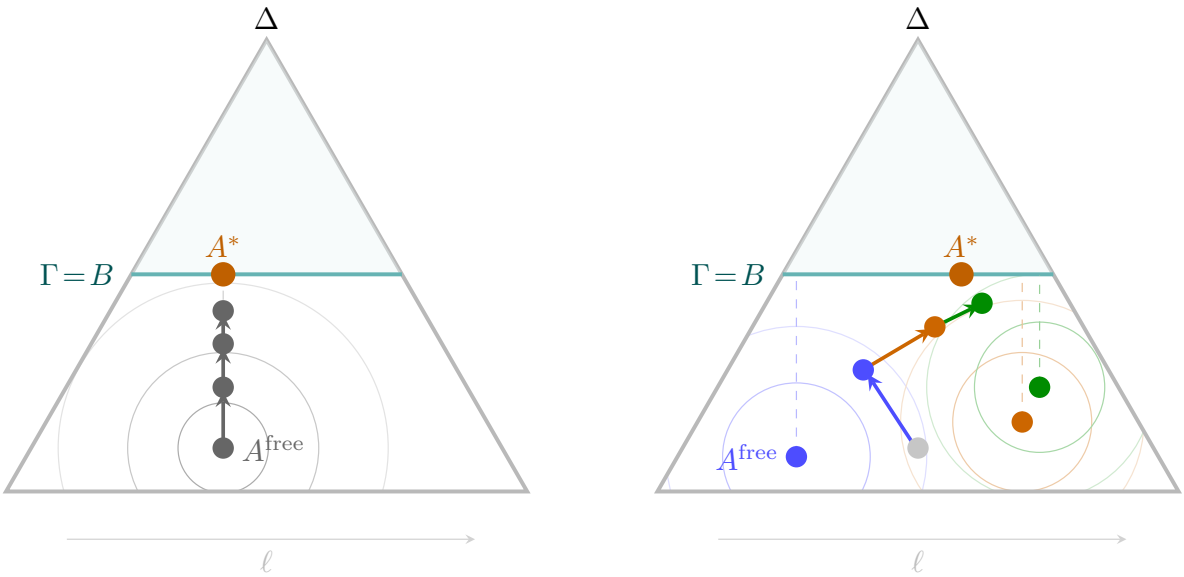
The local guarantee that holds at each layer is: given $s^{(\ell)}$ and $\Lambda^{(\ell)}$, the attention $A^{(\ell)}$ is the exact minimum-KL solution for *that layer's* primal (Proposition 2-3.3). What is not guaranteed is that $\Lambda^{(\ell)}$ equals the optimal dual for layer ℓ's problem; it is instead the accumulated sum of deficits from all previous layers, each formed under different content scores.

The two regimes were previewed in Figure 2-5 (introduced in Section 2-1). Panel (a) shows the stationary case where A^{free} is fixed and the trajectory converges monotonically to A^*. Panel (b) shows the non-stationary case: A^{free} shifts at every layer, changing the KL landscape and causing the dual update direction to change sign between layers. The algorithm carries $\Lambda^{(\ell)}$ forward as a warm start rather than iterating to convergence on any single problem.

The implicit assumption is *cross-layer stationarity of violations*: a pair that was structurally under-attended in layer $\ell-1$ tends to remain under-attended in layer ℓ, so the carried-forward dual exerts useful structural pressure even though it was computed under an earlier content distribution. When this assumption holds, the algorithm acts as an online controller that increases structural pressure where deficits persist and relaxes it where they do not. When it fails (for instance, if the model learns to resolve a deficit in one layer and reintroduce it in the next), the dual may oscillate rather than accumulate. The dual magnitude dynamics in Section 3 provide an empirical check on whether violations persist in practice.

2-4-3 Comparison to Fixed Logit Bias

A fixed bias uses one global scale α for every graph, every layer, and every pair, so the logit correction is $\alpha\, f_{hij}$ regardless of whether content attention already agrees with the structural prior. Constraint attention applies $\Lambda_{ij}^{(\ell)}\, f_{hij}$ with Λ set per-pair and per-layer by the running budget deficit. Pairs that already satisfy their budget do not accumulate dual pressure; pairs that persistently violate the budget do.



(a) stationary **(b) non-stationary**

Figure 2-5: Optimisation trajectories in the probability simplex Δ. *Panel (a)*: A^{free} fixed; the trajectory converges to the minimum-distortion solution A^* on the constraint hyperplane $\Gamma = B$. *Panel (b)*: A^{free} shifts each layer; the update direction changes sign between layers, producing the non-stationary regime of constraint attention in practice.

The detachment of Λ from the gradient graph has a further consequence: content logits cannot silently cancel the structural term. In a fixed-bias model, gradient descent can learn $s_{hij} \approx -\alpha f_{hij}$, nullifying the structural correction. In constraint attention, any such cancellation produces a persistent deficit in Γ relative to B, which causes Λ to grow on the next layer until the cancellation is overcome. The structural requirement is enforced analytically, independent of what gradient descent does to the content logits or to f.

Proposition 2-3.3 makes this precise: at the layer where content and dual reach their fixed point, the realised attention is the minimum-KL projection of the content-only distribution onto the structural constraint set. A fixed bias cannot achieve this optimum because it applies the same scale to every pair and every layer, over-correcting where content already agrees with the prior and under-correcting where the deficit is large.

2-5 Constraint Attention Algorithm and Forward Pass

This chapter gives the computational form of the method introduced in Chapter 2. The constraint framework requires a pairwise structural score $f_{hij}^{(k)}$ as input; this chapter describes the two main families of such scores explored in this thesis.

The first is the **Graphormer spatial encoding** [34]: a learned lookup table indexed by shortest-path distance that maps the discrete graph metric into a per-head attention logit. This encoding is interpretable, parameter-efficient, and already in logit space.

The second is the **RRWP embedding** introduced by GRIT [16]: a learned linear map from multi-hop random-walk landing probabilities into per-head attention logits. RRWP captures a continuous, multi-scale view of structural proximity that complements the discrete distance metric of the SPD encoding.

Section 2-5-1 describes both families and explains the choice of each per dataset. Section 2-5-2 gives the forward-pass algorithm in its minimal form and analyses its cost; the full multi-signal version is given in Section A-4 of Appendix A, and full pipeline details for the structural signals are in Appendix B.

2-5-1 Pairwise Structural Signals

Before any transformer layer runs, every pairwise structural signal must be converted into a structural score $f_{hij}^{(k)}$ that feeds the budget computation in Algorithm 1. This thesis uses two families of signal, corresponding to the two dominant approaches in the graph-transformer literature [34, 18, 16]. Full pipeline details are given in Appendix B.

Family 1: Graphormer Spatial Encoding (SPD + bond type). Graphormer [34] encodes graph structure as a learned scalar bias indexed by the shortest-path distance between each pair of nodes: $f_{hij}^{(\mathrm{SPD})} = e_{\mathrm{SPD}(i,j)}$ for a lookup table $e \in \mathbb{R}^{D_{\max}}$. This encoding lives directly in attention-logit space—no normalisation is needed to use it as a budget score—and assigns a single shared logit to all pairs at the same graph distance. A bond-type encoding complements it by embedding the discrete chemical bond category of each edge into a per-head logit, providing local chemical specificity that the distance-only SPD lacks. Together, SPD and bond type are the primary structural signal used on ZINC-12k. Figure 2-6 shows the SPD pipeline.

Family 2: RRWP Embedding. GRIT [16] introduced the use of Relative Random-Walk Probabilities (RRWP) as pairwise attention bias. For each pair (i, j), RRWP collects the k-step random-walk landing probabilities $R_{ij}^{(k)}$ for $k = 0, \ldots, T-1$, where $R_{ij}^{(k)}$ is the probability that a uniform random walk leaving node i sits at node j after exactly k steps. These probabilities are the entries of the k-th power of the row-normalised random-walk transition matrix (built from the graph adjacency and node degrees; defined in full in Appendix B-3). Stacked over k, they form an edge-level feature vector $\mathbf{E}_{ij} \in [0, 1]^T$, which a learned linear map projects to per-head logits. Unlike SPD, which assigns the same logit to all pairs at a given graph distance, RRWP weights pairs by path multiplicity and local degree, capturing

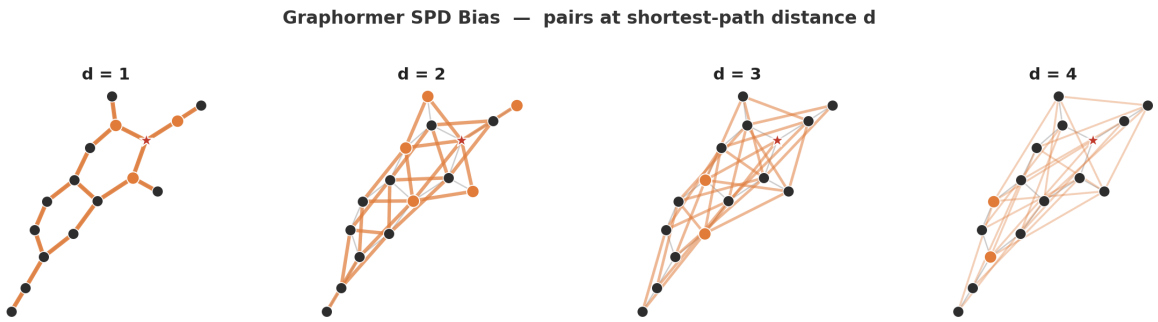


Figure 2-6: SPD bias pipeline: shortest-path distances are mapped through a learned lookup table to produce a per-head scalar logit $f_{hij}^{(\mathrm{SPD})}$. After row-wise softmax the result forms the structural budget $B_{ij}^{(\mathrm{SPD})}$, assigning above-uniform probability to pairs close by the graph metric.

a continuous, multi-scale view of structural proximity. RRWP is the primary structural signal used on the Peptides benchmarks, where the longer-range and more varied connectivity makes the richer multi-hop representation beneficial. Figure 2-7 illustrates the random-walk spreading.

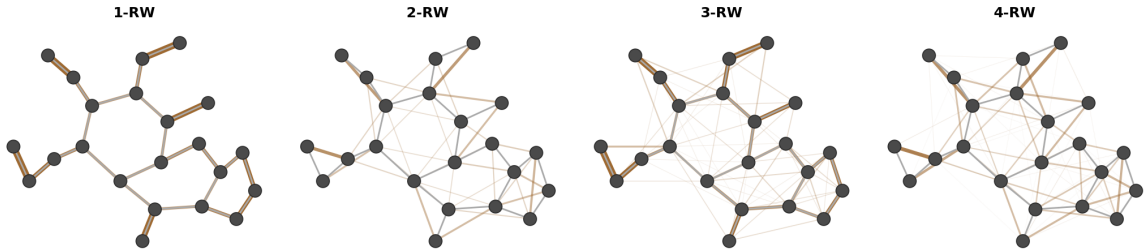


Figure 2-7: Random walk spreading from a query node at 1-, 2-, 3-, and 4-hop distances on a molecular graph. Stacking all T hop-matrices into $\mathbf{E}_{ij}$ gives a graded, multi-scale view of structural proximity between every pair of atoms.

2-5-2 Forward Pass Algorithm

Constraint attention changes a fixed-bias structural transformer in exactly one place. A Graphormer-style layer adds a fixed structural term to the attention logit, $A_{hij} = \mathrm{softmax}_j(s_{hij} + f_{hij})$, and uses the same term at every layer. Constraint attention scales that term by a per-pair dual variable Λ_{ij} and, after each layer, nudges the dual by the budget deficit $B_{ij} - \Gamma_{ij}$. Algorithm 1 shows this minimal form for a single structural signal.

The attention interface is untouched: each layer is still a row-wise softmax over keys, and the transformer update (value projection, residual, layer norm, FFN) is exactly that of the base model. The single structural term f_{hij} is replaced by the dual-scaled term $\Lambda_{ij} f_{hij}$, and one element-wise update of Λ (line 7) is added after each layer. Setting $\Lambda^{(0)} = 1$ and $\eta = 0$ freezes the dual and recovers a fixed Graphormer-style bias exactly.

The full algorithm (multiple structural signals, the detached dual, the explicit transformer update, and the warm-start/reset behaviour across forward passes) is given in Section A-4 of

Algorithm 1 Constraint Attention (minimal form, single signal)

Require: structural score f_{hij} and budget B_{ij} (Appendix B); dual step η, warm start $\Lambda^{(0)}$

$\Lambda_{ij} \leftarrow \Lambda^{(0)}$ ▷ $\Lambda^{(0)}$=1, η=0 is Graphormer
for each layer $\ell = 0, \ldots, L-1$ **do**
  $s_{hij} \leftarrow (W_Q^{(h)} x_i^{(\ell)})^\top (W_K^{(h)} x_j^{(\ell)})/\sqrt{d_k}$ ▷ content score (standard attention)
  $A_{hij} \leftarrow \mathrm{softmax}_j(s_{hij} + \Lambda_{ij}\, f_{hij})$ ▷ Graphormer logit, f scaled by the dual
  $x^{(\ell+1)} \leftarrow \mathrm{TransformerUpdate}(x^{(\ell)}, A)$ ▷ unchanged
  $\Gamma_{ij} \leftarrow \frac{1}{H} \sum_h A_{hij}$ ▷ head-averaged attention
  $\Lambda_{ij} \leftarrow \left[\Lambda_{ij} + \eta\,(B_{ij} - \Gamma_{ij})\right]_+$ ▷ the only new step
end for

Appendix A, which also lists the hyperparameters η and $\Lambda^{(0)}$ and the per-dataset choice of structural signal.

Remark 2-5.1 (Computational overhead and memory). The structural scores and budgets are computed once per forward pass before the layer loop; their cost is amortised over all L layers. Inside the loop, the dual update adds $O(KN^2)$ element-wise operations per layer: one subtraction and one clamp per pair per signal. With $K = 1$ or $K = 2$ signals, this is a scalar operation on an $N \times N$ matrix, which is negligible compared to the $O(N^2 d_k)$ matrix multiplications of the standard attention computation: for $d_k = 64$ and $K = 2$, the dual overhead is roughly $2/64 \approx 3\%$ of the attention cost per layer. Across a 10-layer forward pass the total additional arithmetic from the dual mechanism is less than one attention layer.

The memory overhead is $O(KN^2)$ scalars for the dual state, compared to $O(N^2 H)$ for the full attention weight matrices. For the ZINC-12k configuration ($N \leq 64$, $H = 8$, $K = 2$), the dual state holds $2 \times 64^2 = 8{,}192$ floats, while the attention matrices hold $8 \times 64^2 = 32{,}768$ floats per layer, a factor of 4 less. The dual state is a single persistent matrix shared across all L layers; it does not grow with depth.

Chapter 3

Experiments

3-0-1 Experimental Setup

Experiments are run on three molecular-graph benchmarks:

- **ZINC-12k.** Graph-level regression of penalised logP (the ZINC "constrained solubility" target) on a 10k/1k/1k train/val/test split, evaluated by MAE (lower is better).
- **Peptides-func.** Multi-label classification of 10 peptide function categories from the Long Range Graph Benchmark (LRGB) [11], evaluated by macro average precision (AP, higher is better).
- **Peptides-struct.** Multi-target regression of 11 peptide structural properties from LRGB, evaluated by MAE (lower is better).

The benchmarks differ in graph size ($n \approx 23$ for ZINC, $n \approx 151$ for Peptides) and in the role of structural locality: ZINC graphs are small enough that local topology carries most of the signal, while the LRGB tasks are explicitly designed to require long-range interactions between non-adjacent atoms. This difference motivates the choice of structural prior: ZINC experiments use the Graphormer SPD and bond-type bias, which captures discrete graph-metric proximity and is effective for small, locally structured graphs; Peptides experiments use the learnable RRWP embedding, which captures continuous multi-hop proximity and is better suited for larger graphs where long-range dependencies matter.

Full architecture and training details are given in Appendix A. The central comparison is a tightly matched ON/OFF ablation:

- **ON.** Full method: probability-budget dual update $\Lambda^{(\ell+1)} = [\Lambda^{(\ell)} + \eta(B - \Gamma^{(\ell)})]_+$ at every layer, with $\Lambda^{(0)}$ and η as hyperparameters set per experiment (Appendix A-3).
- **OFF.** Λ is frozen at 1 throughout ($\eta = 0$, $\Lambda^{(0)} = 1$); equivalent to a fixed additive structural logit bias.

Results report mean ± sample standard deviation across three independent runs. All experiments are run on a single rented RTX 3090 GPU.

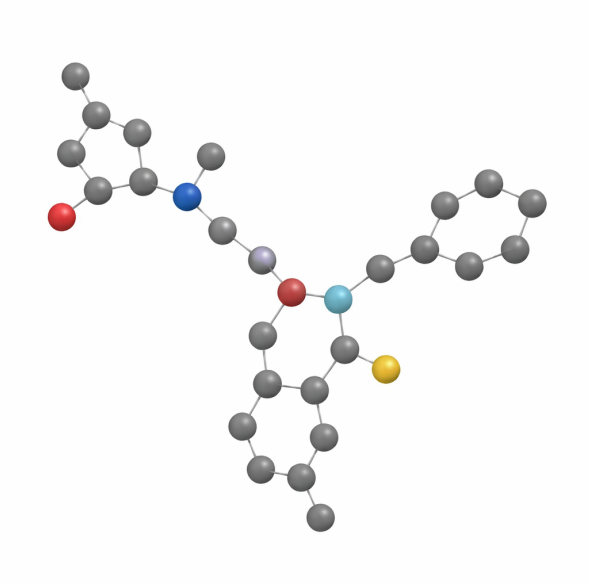


Figure 3-1: Representative molecular-graph topology. Small graphs built from ring substructures and side branches (ZINC, $n \approx 23$) are in a regime where local structural constraints are effective. Peptide graphs ($n \approx 151$) are larger and require long-range interactions that local structural priors may conflict with.

3-0-2 Main Results

ZINC-12k. Table 3-1 compares constraint attention (SPD + bond-type budget, $\Lambda^{(0)} = 2$, $\eta = 3$) to published GNN baselines on ZINC-12k [34].

Method	ZINC-12k MAE ↓
Published baselines [34]	
GIN	0.526
GCN	0.367
PNA	0.142
SAN	0.139
GraphormerSLIM	0.122
SPD + bond budget ($\Lambda^{(0)} = 2$, $\eta = 3$)	
Fixed (OFF)	0.1778 ± 0.0166
Constraint (ON)	**0.1387** ± 0.0066
RRWP budget ($\eta = 7$)	
Fixed (OFF)	0.2088 ± 0.0272
Constraint (ON)	0.1557 ± 0.0193

Table 3-1: ZINC-12k test MAE. "Fixed" is the OFF condition; "Constraint" is the ON condition. Both budget types use three seeds at 1000 epochs; the SPD + bond row uses the sweep-optimal $\Lambda^{(0)} = 2, \eta = 3$, the RRWP-only row uses $\Lambda^{(0)} = 0, \eta = 7$. Results show mean ± std. Baselines from Ying et al. [34].

Constraint attention reduces mean test MAE from 0.178 to 0.139 (-22.0%), winning consis-

tently across runs. All baseline numbers in Table 3-1 are taken from Ying et al. [34]; only the OFF/ON rows are our runs. The remaining gap to GraphormerSLIM (0.122) is accounted for by its use of degree-centrality node encoding, which constraint attention cannot exploit, since adding it deactivates the dual (Section 3-0-4); the structural prior itself is identical.

The RRWP-only rows in Table 3-1 confirm the dual mechanism is not specific to the SPD+bond budget: with the learnable RRWP budget alone (no SPD, no bond bias), ON reduces MAE from 0.209 to 0.156 (-25%), winning on every seed. The dual update therefore separates from the fixed-budget baseline under both structural priors tested on ZINC.

Peptides benchmarks. Table 3-2 compares constraint attention (RRWP budget, $\eta = 0.15$) to LRGB baselines on both Peptides tasks [11].

Method	**Peptides-func AP** ↑	**Peptides-struct MAE** ↓
Published baselines [11]		
GCN	0.5930 ± 0.0023	0.3496 ± 0.0013
GINE	0.5498 ± 0.0079	0.3547 ± 0.0045
GatedGCN	0.5864 ± 0.0035	0.3420 ± 0.0013
GatedGCN+RWSE	0.6069 ± 0.0035	0.3357 ± 0.0006
Transformer+LapPE	0.6326 ± 0.0126	0.2529 ± 0.0016
SAN+LapPE	0.6384 ± 0.0121	0.2683 ± 0.0043
SAN+RWSE	0.6439 ± 0.0075	0.2545 ± 0.0012
GPS	0.6535 ± 0.0041	0.2500 ± 0.0012
GRIT	0.6988 ± 0.0082	0.2460 ± 0.0012
RRWP budget		
Fixed (OFF)	0.6205 ± 0.0102	0.2498 ± 0.0029
Constraint (ON)	**0.6246** ± 0.0095	**0.2482** ± 0.0017

Table 3-2: Peptides-func and Peptides-struct test results. "Fixed" is the OFF condition; "Constraint" is the ON condition (RRWP budget, $\eta = 0.15$). Results show mean ± std across three seeds. Baselines from Dwivedi et al. [11].

Using the RRWP budget, constraint attention improves mean test AP from 0.621 to 0.625 on Peptides-func, a very mild gain of the same order as the seed-to-seed standard deviation. On Peptides-struct, three seeds give 0.248 ± 0.002 (ON) vs 0.250 ± 0.003 (OFF); the mean gap is within one standard deviation of seed-to-seed variation, so the dual mechanism does not detectably separate from the fixed-budget baseline at $\eta = 0.15$ on this task.

3-0-3 Dual Dynamics: How the Constraint Mechanism Operates

At the start of each forward pass Λ is at its warm-start value. On the first layer, content attention is uncorrected and may under-attend structurally important pairs, producing non-zero deficits $B_{ij} - \Gamma_{ij}^{(0)} > 0$. As layers progress, the dual grows for persistently under-attended pairs, increasing the structural logit bias and steering attention toward the budget. For pairs where content attention already respects the budget, the deficit is negative and $[\cdot]_+$ keeps Λ at zero. The dual therefore acts as a selective amplifier: it is active only where the forward pass reveals a structural deficit.

Figure 3-2 shows this dynamic empirically on ZINC-12k under the sweep-optimal $\Lambda^{(0)} = 2$, $\eta = 3$ setting. Each violin captures the full distribution of pairwise $|\lambda|$ at a given layer; the solid line marks the mean and the dashed lines the 90th and 99th percentiles. Two things stand out. First, the mean grows steadily with depth: the average pair accumulates structural pressure rather than reaching a stationary Λ early on. Second, the spread between mean and tail widens with depth, so pairs differentiate from one another as the forward pass progresses. Both observations are consistent with the cross-layer behaviour analysed in Section 2-4: deficits persist long enough across layers for the carried-forward dual to do meaningful work, rather than being resolved within a single layer.

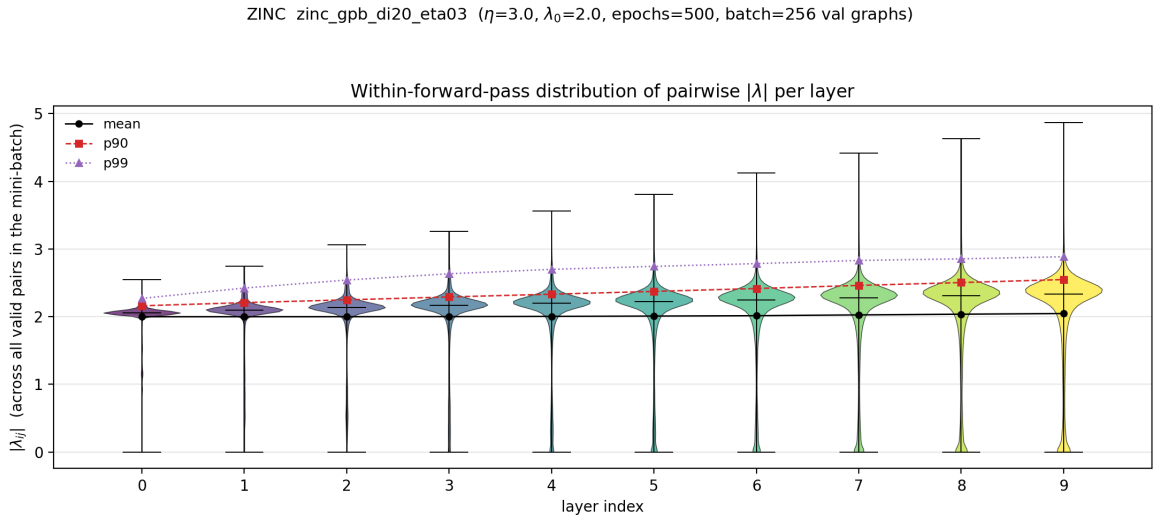


Figure 3-2: Within-forward-pass distribution of pairwise $|\lambda|$ per layer on ZINC-12k graphs ($\Lambda^{(0)} = 2$, $\eta = 3$). Each violin shows the full distribution across all node pairs at that layer; the solid line is the mean, dashed lines mark the 90th and 99th percentiles. The distribution spreads with depth as pairs accumulate differentiated structural pressure.

3-0-4 Limitation: Constraint Deactivation by Shared Embeddings

Constraint attention assumes that the structural score f_{ij} and the content scores s_{hij} are independent: f_{ij} is derived from graph topology, while s_{hij} is driven by learned node-feature projections. This independence is what allows the dual to learn a meaningful correction: the logit is $s_{hij} + \Lambda_{ij}\, f_{ij}$, and since f_{ij} also defines the budget via $B_{ij} = \mathrm{softmax}_j(f_{ij})$, the deficit $B_{ij} - \Gamma_{ij}$ measures how far content attention falls short of the structural target.

When node features encode structural information (here, degree centrality $z_{\deg(v_i)}$, defined in Appendix B-4), the content scores s_{hij} learn to approximate f_{ij}, because the queries and keys now carry degree information that correlates with the SPD/bond budget. The logit then behaves as $s_{hij} + \Lambda_{ij} f_{ij} \approx (\alpha + \Lambda_{ij}) f_{ij}$ for some learned scalar α, so $\Gamma_{ij} \approx B_{ij}$ regardless of Λ_{ij}. The deficit collapses to near zero, the dual cannot grow, and f_{ij} can no longer be learned as an independent corrective bias: it is *tied* to the budget that the content attention already satisfies.

The centrality encoding experiment confirms this effect empirically. Adding $z_{\deg(v_i)}$ to the

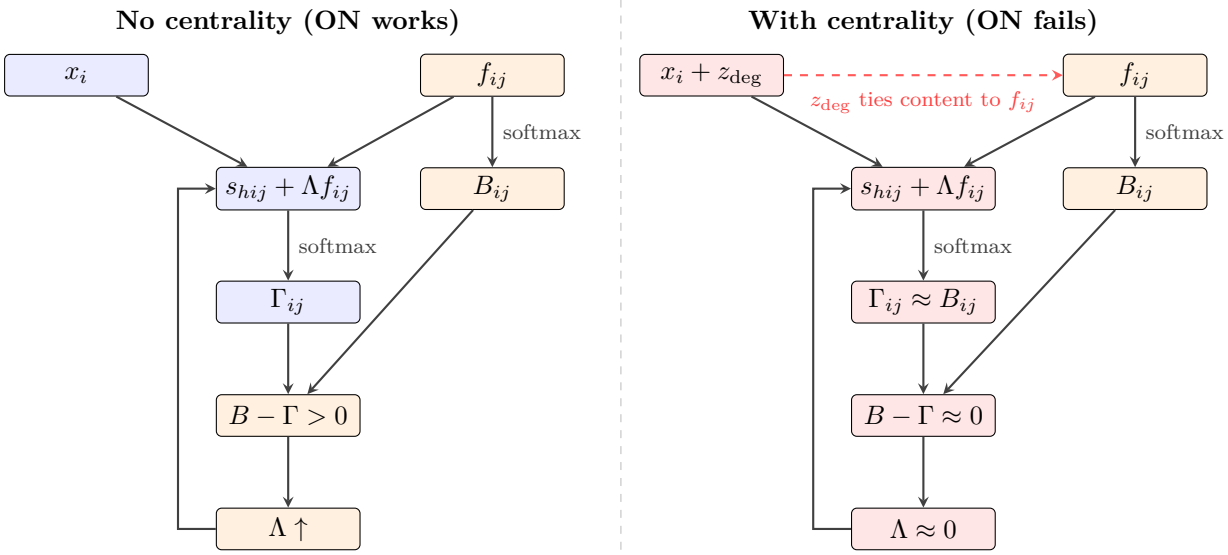


Figure 3-3: Why centrality deactivates the dual. *Left:* f_{ij} is the structural score; it feeds both the budget $B_{ij} = \text{softmax}(f_{ij})$ and the logit correction $\Lambda_{ij} f_{ij}$, independently of the content scores s_{hij}. The deficit $B - \Gamma$ is non-trivial and Λ grows. *Right:* The centrality embedding z_{deg} encodes the same structural information as f_{ij}, so s_{hij} already aligns with the budget. The attention $\Gamma_{ij} \approx B_{ij}$ regardless of Λ, the deficit collapses, and f_{ij} can no longer serve as an independent corrective bias.

ZINC-12k model dramatically improves the fixed-prior baseline while simultaneously collapsing the dual ($|\lambda| \to 0.35$). Constraint attention then underperforms the fixed prior.

Method	Centrality	Test MAE
Fixed (OFF)	no	0.1778 ± 0.0166
Constraint (ON)	no	$\mathbf{0.1387} \pm 0.0066$
Fixed (OFF)	yes	**0.1341**
Constraint (ON)	yes	0.1612

Table 3-3: Effect of degree centrality on the ZINC-12k ON/OFF gap. "No centrality" rows are the main SPD+bond results from Table 3-1 (test MAE, mean $\pm$ std across three seeds). "Centrality" rows are best-val MAE from single-seed runs with degree centrality added to the node features. Without centrality ON wins; with centrality, OFF improves dramatically and ON loses to OFF.

3-0-5 Ablation: Hyperparameter Sensitivity

Constraint attention introduces only two hyperparameters beyond the base transformer: the dual initialisation $\Lambda^{(0)}$ and the dual step size η. Importantly, two corner cases recover existing baselines exactly. Setting $\eta = 0$ (no update) with any $\Lambda^{(0)} > 0$ recovers the standard fixed-bias model; setting $\Lambda^{(0)} = 0$ and $\eta = 0$ recovers plain content-only attention. The adaptive constraint mechanism therefore generalises both without additional architecture changes.

Table 3-4 shows the full ZINC-12k test MAE grid across all combinations of $\Lambda^{(0)} \in \{0.0, 0.5, 1.0, 2.0, 3.0\}$ and $\eta \in \{0, 1, 3, 7, 10\}$, each a single-seed run of 500 epochs. Because every cell is a single seed, individual entries carry seed-level noise. Only the larger gaps in the grid should be read as significant; small cell-to-cell differences within the adaptive region are within the expected seed-to-seed spread.

	Dual initialisation $\Lambda^{(0)}$				
Step η	0.0	0.5	1.0 (Graphormer)	2.0	3.0
0 (fixed)	0.690	0.223	0.192	0.199	0.214
1	0.198	0.188	0.168	0.160	0.158
3	0.186	0.188	0.142	**0.127**	0.137
7	0.165	0.163	0.151	0.148	0.142
10	0.163	0.175	0.160	0.154	0.150

Table 3-4: ZINC-12k test MAE across dual step size η and initialisation $\Lambda^{(0)}$, each a single-seed 500-epoch run. Greener is better. $\Lambda^{(0)} = 1.0$, $\eta = 0$ recovers the Graphormer fixed-bias baseline; $\Lambda^{(0)} = 0.0$, $\eta = 0$ is content-only attention. Best result in bold.

Two robust patterns emerge from the grid; smaller differences within the adaptive region are not commented on, as they fall within seed-level noise.

Structure is indispensable. Content-only attention ($\Lambda^{(0)} = 0$, $\eta = 0$) achieves MAE 0.690, roughly four to five times worse than every configuration that includes a structural signal. The structural prior, not the adaptive mechanism alone, is the primary driver of performance on this task.

Adaptive updates clearly outperform the fixed bias. The best fixed-bias entry ($\Lambda^{(0)} = 1.0$, $\eta = 0$) reaches 0.192; turning the dual on at the same initialisation reaches 0.142 at $\eta = 3$, and the sweep-optimum $\Lambda^{(0)} = 2.0$, $\eta = 3$ reaches 0.127. The gap between the best fixed and best adaptive cells is large enough to survive single-seed noise.

Running the sweep optimum ($\Lambda^{(0)} = 2$, $\eta = 3$) for three independent seeds at the full 1000-epoch schedule gives 0.139 ± 0.007, confirming that the setting generalises beyond the single-seed sweep.

3-1 Conclusion

This thesis introduced constraint attention, a method that encodes structural priors as per-pair inequality constraints on the attention distribution rather than as fixed additive biases. Each constraint is enforced through a Lagrangian dual variable that is updated layer by layer based on how much the content attention violated the structural budget. This gives the model a principled, optimisation-based mechanism for incorporating graph structure, with the strength of the structural correction determined automatically by constraint violations rather than by a global hyperparameter.

Results. On ZINC-12k, constraint attention with Graphormer-style structural encoding reduces mean test error by more than twenty percent over the equivalent fixed-bias model, winning on every seed across three independent runs. The improvement is specific to the adaptive dual: both settings use the same structural prior, so the gain comes entirely from the per-pair feedback mechanism. On the Peptides benchmarks the learnable RRWP budget matches the fixed-prior baseline within seed-level noise, so while the mechanism transfers beyond the Graphormer encoding, it does not separate clearly from the fixed baseline on these longer-range tasks.

Limitation. The centrality encoding ablation identified the main practical boundary of the method: when the node embeddings already encode enough structural information for content attention to naturally respect the budget, the dual variables stay near zero and the constraint mechanism has nothing to correct. In that setting, adding a fixed structural bias is simpler and equally effective. The dual magnitude during training is a reliable diagnostic for this: low activity signals that a fixed bias suffices.

Future work. The most promising direction for closing the gap to state-of-the-art models is to make the cross-layer connection more expressive. In the current formulation the dual update is a fixed gradient step with a global step size: the update direction is determined solely by the instantaneous deficit, and no information from earlier layers is retained beyond the accumulated dual value. This is a deliberately minimal controller, chosen for its theoretical grounding, but it leaves expressive power unused.

A natural extension is to replace the fixed update rule with a learned one. A small per-pair recurrent module could take the history of structural deficits and the current dual value as input, and learn to predict the update direction and magnitude that best drives convergence for the current graph and layer. This would allow the model to adapt its structural enforcement strategy per instance rather than per dataset, and could learn to handle the non-stationary violations that the fixed step struggles with. A complementary direction is to allow the structural budget itself to be refined across layers, turning the prior from a fixed input into a latent variable that the model sharpens as it builds up a representation of the graph. Together these extensions would transform constraint attention from a fixed optimisation algorithm embedded in a transformer into a learned one, retaining the structural guarantees of the constraint formulation while giving the model the flexibility needed to reach state-of-the-art performance on larger and more varied benchmarks.

Appendix A

Experimental Setup

This appendix gathers the experimental details needed to reproduce the results in Chapter 3: datasets, base architecture, training hyperparameters, and a detailed specification of the forward pass (Section A-4). The budget construction and structural-signal pipelines are given in Appendix B.

A-1 Datasets

ZINC-12k. Graph-level regression of the penalised logP solubility score on a 10k/1k/1k train/val/test split. Graphs have a median of 23 atoms. Evaluated by mean absolute error (MAE, lower is better).

Peptides-func. Multi-label binary classification of 10 peptide functional categories from the Long Range Graph Benchmark (LRGB) [11]. Graphs have a mean of 150.9 nodes and 307.3 edges. Evaluated by macro average precision (AP, higher is better). The benchmark is explicitly designed to require long-range interactions between non-adjacent atoms.

Peptides-struct. Multi-target regression of 11 peptide structural properties from LRGB, using the same graphs as Peptides-func. Evaluated by mean absolute error (MAE, lower is better).

All datasets use the official train/val/test splits. No data augmentation is applied. Each ON/OFF cell is run with three random seeds; results report mean and sample standard deviation across seeds.

A-2 Base Architecture

A dense graph transformer (full attention, no edge masking) is used for all benchmarks. ZINC-12k uses the narrower configuration in Table A-1; Peptides-func and Peptides-struct

use the wider hidden dimension in Table A-2 to match the LRGB parameter budget. Each layer applies multi-head self-attention over all node pairs, followed by a two-layer feed-forward network with ReLU activation, residual connections, and layer normalisation.

Hyperparameter	Value
Hidden dimension d	64
Number of attention heads H	8
Number of transformer layers	10
FFN hidden dimension	128
Total parameters (approx.)	340,545

Table A-1: Architecture for ZINC-12k.

Hyperparameter	Value
Hidden dimension d	80
Number of attention heads H	8
Number of transformer layers	10
FFN hidden dimension	160

Table A-2: Architecture for Peptides benchmarks.

A-3 Full Training Hyperparameters

Hyperparameter	ZINC SPD+bond	ZINC RRWP	Peptides
Dual settings (ON)			
Dual step η	3	7	0.15
Warm-start $\Lambda^{(k,0)}$	2.0	0.0	0.0
Structural signal(s)	SPD + bond	RRWP	RRWP
OFF baseline (all experiments): $\Lambda^{(k,\ell)} \equiv 1$, $\eta = 0$			
Training			
Batch size	32	32	16
Epochs	1000	1000	50
Peak learning rate	10^{-3}	10^{-3}	3×10^{-4}
Attention dropout	0.0	0.0	0.1
Random seeds	3	3	3
Signal-specific			
SPD $D_{\max}$ / bond C+1	20 / 5	—	—
RRWP hops K	—	21	32

Table A-3: Hyperparameters for the main results, by experiment. The dual step η and warm start $\Lambda^{(k,0)}$ are defined in Section A-4. Architecture is shared per dataset and listed in Tables A-1–A-2. The Peptides column covers both Peptides-func and Peptides-struct (identical hyperparameters). OFF runs share the matched ON column's settings, differing only in $\Lambda^{(k,\ell)} \equiv 1$ and $\eta = 0$.

A-4 Detailed Forward Pass

Section 2-5-2 gives constraint attention in its minimal single-signal form. Algorithm 2 gives the full version used in the experiments, with multiple signals and the explicit per-head transformer update. For each active signal k, the pipelines of Appendix B produce the score $f_{hij}^{(k)}$ and head-averaged budget $B_{ij}^{(k)}$ once before the loop; only the dual $\Lambda^{(k)}$ changes during the pass.

Detached dual. Each dual $\Lambda_{ij}^{(k)}$ is a non-negative scalar per pair and signal, reset to the warm start $\Lambda^{(0)}$ at the start of every forward pass and accumulated across layers. It is

detached from the autograd graph: the task loss reaches $f^{(k)}$ only through the augmented logit (line 6), while the dual update (line 15) carries no gradient. This one-way coupling is what makes $f^{(k)}$ identifiable (Section 2-3).

Algorithm 2 Constraint Attention Forward Pass (full, multi-signal)

Require: node features $x^{(0)}$; for each active signal k, score $f_{hij}^{(k)}$ and head-averaged budget $B_{ij}^{(k)}$ (Appendix B); layers L, heads H, dual step η, warm start $\Lambda^{(0)}$

1: $\Lambda_{ij}^{(k)} \leftarrow \Lambda^{(0)}$ for all k, i, j ▷ reset and warm-start the dual
2: **for** $\ell = 0, \dots, L-1$ **do**
3: **for** each head h **do**
4: $q_{hi} = W_Q^{(h)} x_i^{(\ell)},\ k_{hj} = W_K^{(h)} x_j^{(\ell)},\ v_{hj} = W_V^{(h)} x_j^{(\ell)}$ ▷ projections
5: $s_{hij} = q_{hi}^\top k_{hj} / \sqrt{d_k}$ ▷ content score
6: $z_{hij} = s_{hij} + \sum_k \Lambda_{ij}^{(k)} f_{hij}^{(k)}$ ▷ dual-augmented logit (Λ detached)
7: $A_{hij} = \mathrm{softmax}_j(z_{hij})$ ▷ attention weights
8: $y_{hi} = \sum_j A_{hij}\, v_{hj}$ ▷ value aggregation
9: **end for**
10: $u_i = \mathrm{concat}_h(y_{hi})\, W_O$ ▷ combine heads
11: $\tilde{x}_i = \mathrm{LN}(x_i^{(\ell)} + u_i)$ ▷ residual + layer norm
12: $x_i^{(\ell+1)} = \mathrm{LN}(\tilde{x}_i + \mathrm{FFN}(\tilde{x}_i))$ ▷ feed-forward sub-block
13: $\Gamma_{ij} = \frac{1}{H} \sum_h A_{hij}$ ▷ head-averaged attention (shared by all k)
14: **for** each signal k **do**
15: $\Lambda_{ij}^{(k)} \leftarrow \left[\Lambda_{ij}^{(k)} + \eta \left(B_{ij}^{(k)} - \Gamma_{ij}\right)\right]_+$ ▷ projected dual ascent on the budget deficit
16: **end for**
17: **end for**

Ensure: $x^{(L)}$; diagnostics $\{\Lambda^{(k)}\}, \{\Gamma\}$

Lines 4–8 are ordinary per-head attention; the only change from a fixed-bias transformer is the $\sum_k \Lambda_{ij}^{(k)} f_{hij}^{(k)}$ term in line 6. Lines 10–12 are the unchanged encoder block (head concatenation and W_O, residual, layer norm, and the feed-forward sub-block). A single attention matrix is shared by all signals, so Γ_{ij} (line 13) is common to every k; each dual then takes one projected-ascent step on its own budget deficit $B_{ij}^{(k)} - \Gamma_{ij}$ (line 15), clamped to stay non-negative. The compute and memory overhead is analysed in Remark 2-5.1 of Section 2-5-2.

Appendix B

Pairwise Structural Signal Pipelines

Algorithm 1 of Section 2-5-2 takes the pairwise structural score $f_{hij}^{(k)}$ as input but treats its construction as a black box. This appendix details the three pipelines that produce $f_{hij}^{(k)}$ in the experiments: shortest-path distance (SPD), bond type, and RRWP.

Every pipeline follows the same template: it starts from a raw graph quantity (BFS distance, bond category, or random-walk landing probability) and maps it to the per-head, per-pair structural score $f_{hij}^{(k)}$ that Algorithm 1 reads directly. The budget used by the dual update follows from this score by the row-wise softmax of Definition 2-1.2, $b_{hij}^{(k)} = \mathrm{softmax}_j(f_{hij}^{(k)})$; the score is never renormalised back from the budget. The three pipelines below differ only in how the score $f_{hij}^{(k)}$ is constructed.

B-1 Shortest-Path Distance (SPD) Pipeline

The SPD pipeline produces the structural score $f_{hij}^{(\mathrm{SPD})}$ from the integer graph metric in two stages.

Stage 1 — All-pairs shortest-path distances. For graph $G = (V, E)$ with n nodes, the shortest-path distance $\mathrm{SPD}(i, j)$ between every ordered pair (i, j) is computed by breadth-first search from each node. Distances are capped at a maximum value $D_{\max}$; pairs with no connecting path (disconnected components) are assigned the special distance $D_{\max} + 1$. The result is an integer matrix $\mathbf{D} \in \{0, 1, \ldots, D_{\max} + 1\}^{n \times n}$.

Stage 2 — Learned distance embedding. A per-head lookup table $e^{(h)} \in \mathbb{R}^{D_{\max}+2}$ maps each integer distance to a scalar logit:

$$f_{hij}^{(\mathrm{SPD})} = e^{(h)}_{\mathrm{SPD}(i,j)}. \tag{B-1}$$

All entries are initialised to zero and trained end-to-end. The lookup is equivalent to the spatial encoding of Graphormer [34], extended to per-head tables. In total the SPD pipeline contributes $(D_{\max} + 2) \times H = 176$ learnable scalars on ZINC ($D_{\max} = 20$, $H = 8$).

B-2 Bond-Type Edge Pipeline

The bond-type pipeline produces the structural score $f_{hij}^{(\text{bond})}$ from the discrete chemical bond category of each edge, providing a chemically specific local prior that complements the SPD signal on ZINC.

Stage 1 — Bond category. Each ordered pair (i, j) carries one of five categories: 0 = no bond (non-adjacent), 1 = single, 2 = double, 3 = triple, 4 = aromatic, recorded as $t_{ij} \in \{0, \ldots, 4\}$.

Stage 2 — Per-head category embedding. A learned per-head table $E_{\text{bond}} \in \mathbb{R}^{(C+1)\times H}$ maps each category t_{ij} directly to a per-head structural score:

$$f_{hij}^{(\text{bond})} = E_{\text{bond}}[t_{ij}]_h. \tag{B-2}$$

The table is initialised so that category 0 (no bond) starts at -1.0 and the remaining categories start at $+0.5$; all entries train normally from there. The pipeline contributes $(C+1) \times H = 5 \times 8 = 40$ learnable scalars on ZINC.

B-3 RRWP Pipeline

The RRWP pipeline produces the structural score $f_{hij}^{(\text{RRWP})}$ as the learned projection of K-step random-walk landing probabilities.

Stage 1 — Random-walk probabilities. Let $\mathbf{A} \in \{0, 1\}^{n\times n}$ be the adjacency matrix of G ($\mathbf{A}_{ij} = 1$ if $(i, j) \in E$ and 0 otherwise), and let $\mathbf{D} = \text{diag}(d_1, \ldots, d_n)$ be the diagonal degree matrix with $d_i = \deg(i) = \sum_j \mathbf{A}_{ij}$. The row-normalised random-walk transition matrix is

$$\mathbf{M} = \mathbf{D}^{-1}\mathbf{A}, \qquad \mathbf{M}_{ij} = \frac{\mathbf{A}_{ij}}{d_i}, \tag{B-3}$$

i.e. $\mathbf{M}_{ij}$ is the probability of stepping from i to a neighbour j in one uniform random step, so each row of $\mathbf{M}$ sums to one. (The molecular graphs used here are connected, so $d_i > 0$ for every node; in general $\mathbf{D}^{-1}$ is read as the pseudo-inverse, which leaves the rows of isolated nodes at zero.) The k-step landing probability from node i to node j is $R_{ij}^{(k)} = [\mathbf{M}^k]_{ij}$. Stacking $k = 0, \ldots, K-1$ gives the RRWP tensor:

$$\mathbf{E}_{ij} = (R_{ij}^{(0)}, R_{ij}^{(1)}, \ldots, R_{ij}^{(K-1)}) \in [0, 1]^K, \tag{B-4}$$

with $K = 21$ hops on ZINC and $K = 32$ on Peptides (Table A-3). Note $\mathbf{R}^{(0)} = \mathbf{I}$ and each row of $\mathbf{R}^{(k)}$ sums to 1 over reachable nodes.

Stage 2 — Log transform. A clipped log-transform is applied to stabilise the scale:

$$\tilde{E}_{ij}^{(k)} = \max\bigl(\log(R_{ij}^{(k)} + \varepsilon),\ -c\bigr), \qquad \varepsilon = 10^{-9}, \quad c = 20. \tag{B-5}$$

The output lies in $[-20, 0]$. Direct neighbours at degree d receive $\tilde{E}_{ij}^{(1)} = \log(1/d) \in [-2.3, -0.7]$; unreachable pairs saturate at -20. The gap of ≈ 17 units is degree-invariant, giving downstream linear layers a consistent discrimination signal.

Stage 3 — Per-head linear projection. A single linear map $W_b \in \mathbb{R}^{H\times K}$ (no bias) projects the log-RRWP tensor to the per-head structural score:

$$f_{hij}^{(\mathrm{RRWP})} = \bigl(W_b\, \tilde{\mathbf{E}}_{ij}\bigr)_h. \tag{B-6}$$

W_b is initialised with a peak-hop pattern (the column at $k = 1$ is biased toward unit weight; other columns are small) so that the initial budget approximates the 1-hop random-walk signal; all weights then train end-to-end. The pipeline has $K \times H$ learnable scalars: 168 on ZINC (K=21, H=8) and 256 on Peptides (K=32, H=8).

B-4 Degree-Centrality Encoding

The three pipelines above produce *pairwise* signals. The degree-centrality encoding used in the constraint-deactivation ablation of Section 3-0-4 is instead a *node-level* feature, and it is not part of the main experiments. It reproduces the centrality encoding of Graphormer [34].

For each node i, let $\deg(i) = \sum_j \mathbf{A}_{ij}$ be its degree. A single learnable embedding table $Z \in \mathbb{R}^{(d_{\max}+1)\times d}$ maps the (capped) degree to a d-dimensional vector that is added to the node feature before the first transformer layer:

$$x_i^{(0)} \ \leftarrow\ x_i^{(0)} + Z_{\min(\deg(i),\, d_{\max})}, \tag{B-7}$$

where $d_{\max}$ caps the degree index (degrees in the molecular graphs used here are small, so saturation is rare). The table is initialised to zero and trained end-to-end.

Equation (B-7) injects each node's connectivity directly into its content representation. This is exactly the coupling analysed in Section 3-0-4: once degree is present in the node features, the content scores s_{hij} can reproduce the structural budget B_{ij} on their own, the deficit $B_{ij} - \Gamma_{ij}$ collapses, and the dual Λ is driven toward zero.

Bibliography

[1] Syed Adnan Akhtar, Arman Sharifi Kolarijani, and Peyman Mohajerin Esfahani. Learning for control: An inverse optimization approach. *IEEE Control Systems Letters*, 6:187–192, 2022.

[2] Turgay Ayer. Inverse optimization for assessing emerging technologies in breast cancer screening. *Annals of Operations Research*, 230(1):57–85, July 2015.

[3] Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

[4] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[5] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems (NeurIPS)*, 33, 2020.

[6] Timothy C. Y. Chan, Rafid Mahmood, and Ian Yihang Zhu. Inverse optimization: Theory and applications. *Operations Research*, 73(2):1046–1074, 2025.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of NAACL-HLT*, 2019.

[8] Ioannis Dimanidis, Tolga Ok, and Peyman Mohajerin Esfahani. Offline reinforcement learning via inverse optimization. *Transactions on Machine Learning Research*, 2026.

[9] Vijay Prakash Dwivedi, Chaitanya K. Joshi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Benchmarking graph neural networks. *Journal of Machine Learning Research*, 24(43):1–48, 2023.

[10] Vijay Prakash Dwivedi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Graph neural networks with learnable structural and positional representations. In *International Conference on Learning Representations (ICLR)*, 2022.

[11] Vijay Prakash Dwivedi, Ladislav Mudigonda Rampasek, Mikhail Galkin, Ali Parviz, Guy Wolf, Anh Tuan Luu, and Dominique Beaini. Long range graph benchmark. In *Advances in Neural Information Processing Systems (NeurIPS) Track on Datasets and Benchmarks*, 2022.

[12] Benjamin Hoover, Yuchen Liang, Bao Pham, Rameswar Panda, Hendrik Strobelt, Duen Horng Chau, Mohammed J. Zaki, and Dmitry Krotov. Energy transformer. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[13] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589, 2021.

[14] Devin Kreuzer, Dominique Beaini, William L. Hamilton, Vincent Létourneau, and Prudencio Tossou. Rethinking graph transformers with spectral attention. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[15] Chuang Liu, Zelin Yao, Yibing Zhan, Xueqi Ma, Shirui Pan, and Wenbin Hu. Gradformer: Graph transformer with exponential decay. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence (IJCAI)*, 2024.

[16] Liheng Ma, Chen Lin, Derek Lim, Haggai Maron, Jaime Carbonell, Mathias Niepert, Jean Kossaifi, and Anima Anandkumar. Graph inductive biases in transformers without message passing. In *International Conference on Machine Learning*, 2023.

[17] André F. T. Martins and Ramón F. Astudillo. From softmax to sparsemax: A sparse model of attention and multi-label classification. *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, 2016.

[18] Luis Müller, Mikhail Galkin, Christopher Morris, and Ladislav Rampášek. Attending to graph transformers. *Transactions on Machine Learning Research*, 2024.

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

[20] Ali Rad, Khashayar Filom, Darioush Keivan, Peyman Mohajerin Esfahani, and Ehsan Kamalinejad. G2RPO: Geometric GRPO escaping LLMs' reasoning ruts to break the accuracy–entropy trade-off. In *Proceedings of the Forty-third International Conference on Machine Learning*, 2026.

[21] Ali Rad, Khashayar Filom, Darioush Keivan, Peyman Mohajerin Esfahani, and Ehsan Kamalinejad. Rate or fate? RLV$^{\varepsilon}$R: Reinforcement learning with verifiable noisy rewards. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*. PMLR, 2026.

[22] Ladislav Rampášek, Mikhail Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and Dominique Beaini. Recipe for a general, powerful, scalable graph transformer. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

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

[24] Ke Ren, Peyman Mohajerin Esfahani, and Angelos Georghiou. Inverse optimization via learning feasible regions. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pages 51471–51488. PMLR, 2025.

[25] Michael E. Sander, Pierre Ablin, Mathieu Blondel, and Gabriel Peyré. Sinkformers: Transformers with doubly stochastic attention. In *Proceedings of the 25th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2022.

[26] Pedro Zattoni Scroccaro, Bilge Atasoy, and Peyman Mohajerin Esfahani. Learning in inverse optimization: Incenter cost, augmented suboptimality loss, and algorithms. *Operations Research*, 73(5):2661–2679, 2025.

[27] Pedro Zattoni Scroccaro, Piet van Beek, Peyman Mohajerin Esfahani, and Bilge Atasoy. Inverse optimization for routing problems. *Transportation Science*, 59(2):301–321, 2025.

[28] Ahsan Shehzad, Feng Xia, Shagufta Abid, Ciyuan Peng, Shuo Yu, Dongyu Zhang, and Karin Verspoor. Graph transformers: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, page 1–20, 2026.

[29] Jonathan M. Stokes, Kevin Yang, Kyle Swanson, Wengong Jin, Andres Cubillos-Ruiz, Nina M. Donghia, Craig R. MacNair, Shawn French, Lindsey A. Carfrae, Zohar Bloom-Ackerman, et al. A deep learning approach to antibiotic discovery. *Cell*, 180(4):688–702, 2020.

[30] Damian Szklarczyk, Rebecca Kirsch, Mikaela Koutrouli, Katerina Nastou, Farrokh Mehryary, Radja Hachilif, Annika Ledergerber Gable, Tao Fang, Nadezhda T. Doncheva, Sampo Pyysalo, et al. The STRING database in 2023: protein–protein association networks and functional enrichment analyses for any sequenced genome of interest. *Nucleic Acids Research*, 51(D1):D638–D646, 2023.

[31] Yingcong Tan, Daria Terekhov, and Andrew Delong. Learning linear programs from optimal decisions. In *Advances in Neural Information Processing Systems*, volume 33, pages 19738–19749, 2020.

[32] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[33] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive study of graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1):4–24, 2020.

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

[35] Yaodong Yu, Sam Buchanan, Druv Pai, Tianzhe Chu, Ziyang Wu, Shengbang Tong, Hao Bai, Yuexiang Zhai, Benjamin D. Haeffele, and Yi Ma. White-box transformers via sparse rate reduction: Compression is all there is? *Journal of Machine Learning Research*, 25(300):1–128, 2024.

[36] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI Open*, 1:57–81, 2020.

Glossary

Acronyms

Acronym	Definition
3mE	Mechanical, Maritime and Materials Engineering
AP	Average Precision
DCSC	Delft Center for Systems and Control
FFN	Feed-Forward Network
GCN	Graph Convolutional Network
GIN	Graph Isomorphism Network
GNN	Graph Neural Network
GPS	General, Powerful, Scalable Graph Transformer
GRIT	Graph Inductive Biases in Transformers (without message passing)
GT	Graph Transformer
KKT	Karush–Kuhn–Tucker
KL	Kullback–Leibler (divergence)
LapPE	Laplacian Positional Encoding
LN	Layer Normalisation
LRGB	Long Range Graph Benchmark
MAE	Mean Absolute Error
MHA	Multi-Head Attention
PNA	Principal Neighbourhood Aggregation
RRWP	Relative Random-Walk Probability
RWSE	Random-Walk Structural Encoding
SAN	Spectral Attention Networks
SPD	Shortest-Path Distance
TU Delft	Delft University of Technology

Symbols

Symbol	Description	Reference
Indices		
i, j	node indices (query, key)	
h	attention head index, $h = 1, \dots, H$	
ℓ	transformer layer index, $\ell = 0, \dots, L$	
k	structural-signal index, $k = 1, \dots, K$	
n, n_i	number of (valid) keys in a graph / row i	
Architecture constants		
H	number of attention heads per layer	
L	number of transformer layers	
T	number of random-walk hops (RRWP)	Def. 1-3.1
d	hidden (model) dimension	
d_k	per-head key / query dimension	
$W_Q^{(h)}, W_K^{(h)}, W_V^{(h)}$	per-head query / key / value projections	Def. 1-3.3
Attention quantities		
$A_{hij}^{(\ell)}$	realised attention weight	Def. 1-3.3
$s_{hij}^{(\ell)}$	content score, $(W_Q x_i)^\top (W_K x_j)/\sqrt{d_k}$	eq. (1-1)
$z_{hij}^{(\ell)}$	augmented logit, $s_{hij} + \Lambda_{ij} f_{hij}$	eq. (2-8)
A_{hij}^{free}	content-only attention $\text{softmax}_j(s_{hij})$	Prop. 2-3.3
Structural prior and budget		
$f_{hij}^{(k)}$	structural score for signal k	Def. 2-1.2
$b_{hij}^{(k)}$	per-head budget, $\text{softmax}_j(f^{(k)})$	eq. (2-2)
$B_{ij}^{(k)}$	head-averaged budget, $\frac{1}{H}\sum_h b_{hij}^{(k)}$	eq. (2-3)
$\hat{B}_{ij}^{(k)}$	score-weighted target, $\sum_h b_{hij}^{(k)} f_{hij}^{(k)}$	eq. (2-4)
Dual mechanism		
$\Lambda_{ij}^{(k,\ell)}$	Lagrangian dual variable	eq. (2-7)
$\Gamma_{ij}^{(\ell)}$	head-averaged realised attention	Prop. 2-3.2
$\tilde{\Gamma}_{ij}^{(\ell)}$	score-weighted realised attention	eq. (2-12)
η	dual step size (head-averaged form)	eq. (2-15)
$\tilde{\eta}_{ij}$	pair-specific rescaled step, $\eta/(H f_{ij})$	eq. (2-14)
$g(\Lambda)$	score-weighted dual function	eq. (2-10)
$F(\Lambda)$	total KL divergence to budget	Prop. 2-3.4
$\mathcal{L}(A, \Lambda, \mu)$	Lagrangian relaxation of the primal	eq. (2-7)
μ_{hi}	multiplier for the row-simplex constraint	eq. (2-7)
Operators and sets		
Δ_n	probability simplex on $\{1, \dots, n\}$	Lem. 2-1.1
$\mathcal{H}(p)$	Shannon entropy, $-\sum_j p_j \log p_j$	Lem. 2-1.1
$\text{softmax}_j(z)$	row softmax over index j	Def. 1-3.3
$[x]_+$	projection onto the non-negative reals, $\max(x, 0)$	eq. (2-12)
$\text{KL}(P\|Q)$	Kullback–Leibler divergence with P as reference	Sec. 1-3