LAYOUTLATTICE: PAIRWISE ATTENTION BIASES FOR CONTEXTUAL LAYOUT GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

We present a simple, yet powerful mechanism to inject pairwise context into layout generation models via attention biases. We target *relation-conditioned* layout generation (the structure-preserving layout-variation setting), which produces alternative arrangements of page elements while preserving given pairwise relations such as which elements overlap and their reading order. Recent approaches based on transformers and continuous diffusion have advanced the field, yet they struggle to capture the pairwise relationships between page elements. Rather than relying solely on per-element features, our approach augments attention with an $N \times N$ bias matrix that injects pairwise priors (overlap/IoU and, optionally, reading order) computed from a reference layout and supplied as a conditioning signal, where N is the number of page elements. The mechanism is drop-in: binary masks and continuous priors are stacked as channels and applied as additive attention biases to a subset of attention heads, while the remaining heads stay unmasked. Across two regimes (max 25 and max 50 page items), our approach improves both realism and structure, e.g., reducing FID from $\sim$11.6 to $\sim$2.7 in the 50-item regime while increasing Max. IoU. Crucially, supplying the *same* priors as plain per-token inputs instead *degrades* the model, showing that the gains stem from the interface rather than mere access to the priors. We further demonstrate generating variations in different canvas sizes by using a relative-area (RA) representation of canvas and page items. Our results suggest that the $N \times N$ attention map is the natural *representation* for pairwise relational structure: a robust, reusable, parameter-free interface for structure-aware layout generation.

1 INTRODUCTION

Automated layout generation underpins high-impact applications including document layout variation, layout completion and refinement, and UI design. Tasks are typically categorized as *unconditional* (generation from no prior context) or *conditional* (guided by partial attributes such as classes, sizes, positions, or incomplete layouts). Earlier methods used GANs (Goodfellow et al., 2014), VAEs (Kingma & Welling, 2014), and transformers (Vaswani et al., 2017), whereas recent work shifted to diffusion for better generation quality and flexible conditioning. Diffusion models (Ho et al., 2020) for layout generation are typically either *discrete* or *continuous*. Discrete diffusion generally represents features as discrete variables and adds elements incrementally on a blank canvas. Continuous diffusion refines a noisy arrangement into an organized one, improving arrangement over each step of the diffusion process.

Diffusion baselines can underperform in structural metrics (alignment, overlap), especially in unconditional settings. Constraint-optimized training addresses this by adding differentiable aesthetic losses (global/local alignment, overlap) with time-dependent weights while unifying conditional tasks via masked inputs. However, these approaches still depend on learned token interactions to reconstruct pairwise/global relationships—leaving intentional overlaps fragile under variation. Some earlier works—including LACE (Chen et al., 2024) —sought to *eliminate* overlaps by penalizing or removing them during training and post-processing. While this can simplify optimization, it conflicts with real-world layouts where controlled overlaps are an intentional design choice (e.g., badges on images, layered UI widgets).

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

We study **relation-conditioned layout generation**: the structure-preserving layout-variation setting in which, beyond per-element attributes, pairwise relations between elements (which elements overlap, and in what reading order) are given from a source layout and must be preserved in the new arrangement. We argue that such pairwise geometric relations are naturally expressed through the attention mechanism. To this end, we introduce a $N \times N$ *attention bias interface* that injects pairwise and global priors—such as IoU and reading order—as additive biases into a selected subset of heads, while leaving the remaining heads unmasked to preserve unconstrained pattern learning. This simple, general design allows the backbone to reuse standard attention without modification. At its core this is a *representation* choice: we represent pairwise relational structure directly in the $N \times N$ attention map (the space where a transformer already reasons about token pairs) rather than as per-element features, which cannot bind a pairwise value to the element pair it describes. Our experiments make this concrete: supplying the *same* relations as flattened per-token inputs degrades the model, whereas representing them in the attention map yields large gains at zero added parameters, evidence that the representation, not access to the relations, is what matters. Combined with a relative-area (RA) representation, this interface naturally accommodates variable page sizes. Crucially, rather than eliminating overlaps, our approach **preserves and modulates them**, ensuring that essential layering cues are retained rather than discarded.

Most prior works restricted the maximum number of page elements to 25, in part because overlaps became unstable beyond that regime and, without a consistent way to enforce or preserve overlaps, generation quality degraded. With our work, we are able to scale evaluations to layouts with up to 50 page items while maintaining strong structure and realism. In summary, the salient contributions are as follows:

- A head-aware $N \times N$ contextual attention interface, a parameter-free *representation* of pairwise structure in the attention map: a subset of heads receives additive biases from pairwise priors (IoU/reading order), while remaining heads stay unmasked to preserve free pattern learning.
- Graded overlap learning: the IoU bias *routes* (which pairs interact); an overlap-consistency loss supervises *magnitude*, aligning predicted pairwise IoU with ground truth, yielding faithful, controllable overlap structure.
- A relative-area (RA) representation that stabilizes geometry and naturally supports page-size variation.
- Scaling beyond 25 page items to 50 page items, with strong improvements across all conditional generation tasks.
- A same-condition evaluation protocol for relation-conditioned generation: every baseline is also given the identical pairwise condition through the only drop-in route (flattened per-token features), isolating the effect of the *representation* from mere access to the condition.

Beyond layout, head-aware $N \times N$ masked attention could apply wherever pairwise or global priors are useful. To validate this claim, we apply the same mechanism to an entirely different domain: *human body pose prediction* and inject anatomical skeleton priors as attention biases into a transformer baseline. Our approach consistently *outperforms the state-of-the-art baseline* across all prediction horizons on 3DPW dataset (Appendix Q).

2 Related Work

Earlier layout generation explored energy-based objectives grounded in design rules, followed by deep generative approaches: GAN (Goodfellow et al., 2014)/VAEs (Kingma & Welling, 2014) for conditional layouts; autoregressive transformers such as LayoutTransformer (Gupta et al., 2021), VTN (Arroyo et al., 2021), etc. improved quality but were less flexible for conditional tasks; BLT or bidirectional transformers (Kong et al., 2022) eased conditioning but required knowing the element count; encoder–decoder designs as in LayoutFormer++ (Jiang et al., 2023) introduced backtracking and task-specific retraining; content-aware models added image/text features (Zheng et al., 2019; Seol et al., 2024).

Diffusion-based layout generation unified conditions and improved fidelity. PLay (Cheng et al., 2024) adopts guideline-conditioned latent diffusion using lines to align elements; discrete diffusion models such as LayoutDM (Inoue et al., 2023) and LDGM (Hui et al., 2023) used attribute-specific

corruption (mask-and-replace) and discretized coordinate noise; LayoutDiffusion (Zheng et al., 2023) pushed scale with larger backbones to improve alignment/overlap. LACE (Chen et al., 2024) employs *continuous* diffusion, estimates $\tilde{\mathbf{x}}_0$ by reparameterization, and applies differentiable aesthetic constraints (alignment; overlap) with time-dependent weights, unifying conditional tasks via masked inputs. These backbones encode elements as tokens with positional geometry and rely on learned attention to capture relations.

Relational attention and pair representations. The additive-bias *primitive* is standard; our contribution is *what is injected and how*. Prior biases are either **hand-designed but position-only** (ALiBi (Press et al., 2022): a fixed, content-agnostic distance offset—content matching stays in the learned $QK^\top$) or **learned** (T5 (Raffel et al., 2020), Graphormer (Ying et al., 2021)). We instead inject the **raw, instance-specific pairwise feature** (IoU/RO) directly—a head-start, not a blank slate—with hard-masking (-10^4) forcing zero attention on non-related pairs, while **free heads** keep standard learned attention. More precisely, relational modules that could read pairwise features—GNN/GAT (Veličković et al., 2018), edge-augmented transformers (EGT (Hussain et al., 2022)), Graphormer, relation-aware attention (Shaw et al., 2018)—are all **learned**, and fail our use in one of two ways: the position/distance biases (T5, Shaw, ALiBi, Graphormer's shortest-path term) *generate* their $N \times N$ from **discrete position**, so they cannot represent a per-instance content value like IoU; the edge-feature methods (EGT, GNN) *accept* an arbitrary $N \times N$ matrix but only *digest it through trained layers* (learned projections, soft gates that never reach exactly 0, and—EGT—a heavy per-layer $N \times N \times d$ edge tensor). The structure-prediction lineage (AlphaFold2-style pair representations with triangular attention (Jumper et al., 2021)) is the richest such version—a **learned** pair stack updated every block at up to $O(N^3)$ cost—i.e., the same intuition with *maximal* learned machinery, the opposite end from our unlearned, zero-cost bias. Using a learned module to inject a **known** prior re-learns the structure the prior already specifies (forfeiting the head-start), adds trainable machinery to a deliberately small $\sim$3.7M-parameter backbone, and gives no per-step guarantee the prior is honored. Reduce any of them to our requirement—raw value, hard mask, zero parameters—and it collapses back to our additive bias. We do not claim relational conditioning is novel, nor that no relational module could be trained to help—only that none offers an unlearned, hard-maskable, per-head, zero-parameter injection of a known continuous prior.

Our approach. In practice, most prior works evaluated with at most 25 page items; beyond this, overlap handling often became unstable and structural quality degraded. These approaches underscore a common limitation: capturing inter-dependencies between page items remains challenging. Our approach complements prior arts: explicit priors (IoU, reading order) are injected in attention layer for a selected subset of heads, while remaining heads learn freely; differentiable constraints continue to shape objectives; and a relative-area (RA) representation stabilizes geometry and supports page-size variation. Together, these yield stronger preservation and modulation of overlaps and improved structural metrics without scaling backbone capacity.

3 METHOD

3.1 PROBLEM SETTING: RELATION-CONDITIONED LAYOUT GENERATION

A layout is a set of N elements $\mathbf{x} = \{(c_i, b_i)\}_{i=1}^N$ with class c_i and box b_i. Given a *reference* layout $\mathbf{x}^{\text{ref}}$, we extract a relational condition

$$\mathcal{R}(\mathbf{x}^{\text{ref}}) = \{C_{\text{iou}}, C_{\text{ro}}\}, \qquad C_{\text{iou}}[i,j] = \text{IoU}(b_i^{\text{ref}}, b_j^{\text{ref}}), \quad C_{\text{ro}}[i,j] \in \{0,1\},$$

where C_{ro} marks reading-order-adjacent pairs (C_{ro} is optional). The model generates boxes $\hat{b}_{1:N}$ (and, for completion, missing classes) conditioned on the task's per-element inputs *and* $\mathcal{R}$. Table 1 lists exactly what is given in each task. The condition is a *fixed* input: it is computed once from $\mathbf{x}^{\text{ref}}$, never recomputed from the noisy state or from $\tilde{\mathbf{x}}_0$, and involves no learned predictor.

Where $\mathbf{x}^{\text{ref}}$ comes from. For benchmarking, $\mathbf{x}^{\text{ref}}$ is the test layout itself, exactly as the standard tasks already supply the test layout's own classes (and sizes) (see §4.1).

The central question is representation. Given $\mathcal{R}$, how should a transformer consume it? An $N \times N$ relation can be (a) flattened into per-element features (row i of C appended to token i), or (b) placed in the $N \times N$ attention map, where entry $[i,j]$ is by construction attached to the pair (i,j). We adopt (b) (§3.3) and compare it directly against (a) under the identical condition (§4.2).

Table 1: What is given and what is generated. Standard tasks (Rahman et al., 2021) (left name) become relation-conditioned tasks (right name) by adding $\mathcal{R}(\mathbf{x}^{\text{ref}})$.

Task	Per-element input	Pairwise input	Generated
C→S+P ⇒ C+IoU(+RO)→S+P	classes	C_{iou} ($+C_{\text{ro}}$)	sizes, positions
C+S→P ⇒ C+S+IoU(+RO)→P	classes, sizes	C_{iou} ($+C_{\text{ro}}$)	positions
Completion ⇒ Completion+IoU(+RO)	a subset of elements	C_{iou} ($+C_{\text{ro}}$)	remaining elements

3.2 Preliminaries

Continuous diffusion. We build on the standard DDPM continuous diffusion framework (Ho et al., 2020), in which a clean signal $\mathbf{x}_0$ is gradually corrupted by Gaussian noise and a network learns to reverse the process via a simplified noise-prediction objective $\mathcal{L}_{\text{simple}}$ (Eq. 9). We use the standard $\mathbf{x}_0$-reparameterization to obtain $\tilde{\mathbf{x}}_0$ for geometric losses; see Appendix B for the full derivation.

Multi-head attention. For each of H heads, attention scores are computed as scaled dot-products between Q and K, with an additive bias $\mathbf{B}_h$:

$$\mathbf{S}_h = \frac{\mathbf{Q}_h \mathbf{K}_h^\top}{\sqrt{d_k}} + \mathbf{B}_h. \tag{1}$$

The bias term $\mathbf{B}_h \in \mathbb{R}^{N\times N}$ modulates pairwise attention logits—this is the interface through which we inject contextual priors. The full multi-head attention formulation is given in Appendix C.

3.3 Contextual Attention Masks

Given a batch of B layouts with up to N tokens and padding matrix $S = vv^\top \in \{0,1\}^{N\times N}$, where $v \in \{0,1\}^N$ indicates which of the N elements are real (non-padded), S zeros out any entry involving a padded element. We assemble an attention bias tensor M of shape $[B, H, N, N]$ as follows.

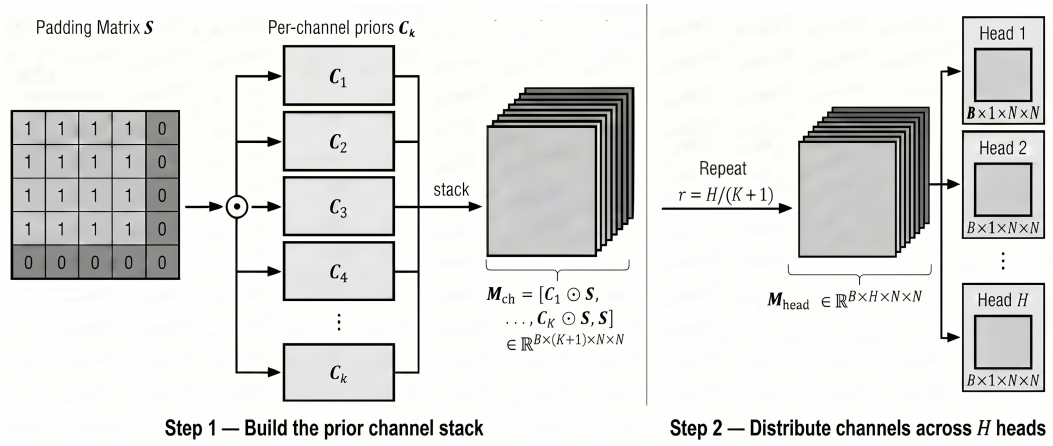


Figure 1: Contextual mask construction: channel assembly

Step 1 — Build the prior channel stack

Gate each prior matrix C_k element-wise by the padding matrix S, then stack the results along the channel dimension, appending S itself as a final channel:

$$M_{\text{ch}} = \left[C_1 \odot S, \ldots, C_K \odot S, S\right] \in \mathbb{R}^{B\times(K+1)\times N\times N}$$

In our experiments we use both $K = 1$ and $K = 2$:

- **IoU matrix (C_{iou})**: $C_{\text{iou}}[i,j] = \text{IoU}(b_i, b_j)$ computed from the normalized boxes of the reference layout $\mathbf{x}^{\text{ref}}$ (§3.1) and supplied as a fixed conditioning input at both training and inference; the diagonal is set to 1.

- **Reading-order matrix (C_{ro})**: binary adjacency (1 for adjacent or linked pairs, 0 otherwise) from the reading order of the reference layout, estimated with an XY-cut–style algorithm (Meunier, 2005).

Additional pairwise channels (e.g., group adjacency, hierarchical proximity, radial distance) can be incorporated by computing their $N \times N$ matrices and appending them to the stack.

Step 2 — Distribute channels across H heads

Let $r = \lfloor H/(K+1) \rfloor$. Tile M_{ch} along the channel dimension r times; assign any remaining heads to S:

$$M_{\text{head}} \in \mathbb{R}^{B \times H \times N \times N}$$

Heads assigned to $C_k \odot S$ are biased by prior k. "Free" heads assigned to S alone are validity-gated but otherwise unconstrained, free to learn arbitrary pairwise patterns among non-padded tokens (head partitioning: Appendix C).

Step 3 — Map zeros to large negatives

$$\hat{M} = \text{where}\big(M_{\text{head}} = 0,\ -10^4,\ M_{\text{head}}\big)$$

Every zero entry in M_{head} is replaced by a large negative ($\approx -\infty$), suppressing two classes of pairs: (i) pairs involving padded tokens, zeroed by S, and (ii) pairs with no prior relation (e.g., IoU $= 0$, non-adjacent reading-order pairs). Non-zero entries are passed through unchanged and serve as additive biases in the attention score. We finally get $\hat{M}$ which is the bias tensor.

Step 4 — Add bias to attention logits

$$S_h = \frac{Q_h K_h^\top}{\sqrt{d_k}} + \hat{M}[:, h, :, :] \qquad (h = 1, \ldots, H) \tag{2}$$

The trainable parameters are the standard transformer weights (query/key/value and output projections, feed-forward layers, and norms) whereas the bias tensor $\hat{M}$ - assembled from priors and not learned - is an additive bias (float mask). Entries set to -10^4 are a numerically stable approximation of $-\infty$ under mixed precision; positive entries (e.g., IoU values or 1 from S) increase the relative weight of permitted pairs. After softmax, entries at -10^4 receive effectively zero attention (Appendix C). The result is head-specific sparsity: an IoU head attends only to overlapping items (IoU > 0), an RO head attends only to allowed/adjacent items, and a group head attends only within groups. **Why $N \times N$ matters (a representation argument):** local features $(x, y, w, h, \text{class})$ are a *per-element* representation and cannot directly encode pairwise interactions such as "element i overlaps j by α," "i precedes j in reading order," or "i and j belong to the same group." Flattening the $N \times N$ relations into per-element inputs does not help: the values are present but a per-token encoder has no way to bind entry $[i, j]$ to the element pair it describes (see the flatten control, Table 2). The $N \times N$ attention map is the right *representation* for these relations—it is indexed by element pairs $[i, j]$, precisely matching the structure of the signal. With M we expose these interactions explicitly in that space. Overlap (IoU) becomes a first-class signal at attention time, enabling the model to preserve and modulate overlaps during variation instead of collapsing them. In complex, real-world pages where overlaps are deliberate (e.g., badges on images, sticky headers), this global context is essential.

Beyond sparsity, the bias *routes* (which pairs interact); the loss supervises *magnitude*. Concretely, we trained with an overlap-consistency loss that penalizes the deviation between the predicted pairwise IoU matrix (computed from $\tilde{\mathbf{x}}_0$) and the ground-truth IoU. Values near zero induce negligible cross-attention and yield little/no overlap in the final layout, whereas larger IoU values manifest as greater, controlled overlap. The components are useful only together (Appendix N): graded overlap is loss-supervised—Figure 2 shows the routing/sparsity, the magnitude comes from the loss. We highlight this as a core contribution of our approach and corroborate it in both 50- and 25-item ablations.

3.4 Relative-Area Representation and Page Token

We encode boxes as $(x_{\text{norm}}, y_{\text{norm}}, h_{\text{norm}}, \sqrt{a_{\text{norm}}})$ with all coordinates normalized by $\max(W, H)$ of the page (consistent with the page token), and $s = \sqrt{a}$ the square root of normalized area a. At

evaluation, width is reconstructed as $w = s^2/h = a/h$ and clamped. This removes one free degree of freedom by deriving w from (h, a), reducing target redundancy.

$$b = (x, y, h, s), \quad s = \sqrt{a}, \quad w = \text{clamp}\left(\frac{s^2}{h}, 0, 1\right) \tag{3}$$

Using height as the primary scale stabilizes targets; width is consistently derived from area (alternatives: Appendix D).

Page token for aspect-ratio awareness: We explicitly add a page token whose normalized encoding depends on the canvas size, so different aspect ratios map to different but comparable embeddings. For example, we append

$$b_{\text{page}} = \left(\frac{W/2}{\max(W,H)}, \frac{H/2}{\max(W,H)}, \frac{H}{\max(W,H)}, 1\right) \tag{4}$$

where the last coordinate denotes full-page area ($\sqrt{a} = 1$) in the RA parameterization. Coupled with RA for items and preservation of reading-order links under transforms, this makes the model page-relative and aspect-ratio aware, enabling generation/refinement across varying page sizes without architectural changes. The contextual masks remain valid under re-scaling because they are computed from normalized geometry. Inference practice is detailed in Appendix D.

3.5 Training Loss

Diffusion and reconstruction loss: as in LACE, we use MSE on token logits and geometry; we apply Min-SNR (Hang et al., 2023) based weighting to both the diffusion loss and the reconstruction loss for timestep balance. Diffusion loss is $\mathcal{L}_{\text{simple}}$ (9), whereas reconstruction loss here is $\mathcal{L}_{\text{rec}}$ = MSE($\tilde{\mathbf{x}}_0, \mathbf{x}_0$). See Appendix F for a detailed explanation of why each loss is weighted the way it is.

Alignment loss (as in LACE): a global alignment loss that uses a binary alignment mask over six standard alignments (left, horizontal center, right, top, vertical center, bottom) to select aligned pairs from coordinate-difference matrices (full definition: Appendix E).

Overlap consistency Loss: MSE between the predicted pairwise IoU matrix from $\tilde{\mathbf{x}}_0$ and ground-truth IoU, evaluated only on valid tokens via masks.

Let $I(x) \in [0,1]^{N\times N}$ be the pairwise IoU matrix computed from boxes in layout x, and let $S = v\,v^\top$ be the valid-pair gate from Section 3.3 (optionally with zeros on the diagonal to ignore self-pairs). We supervise overlaps by matching the predicted clean state $\tilde{\mathbf{x}}_0$ to the ground truth x:

$$\mathcal{L}_{\text{overlap}} = \frac{1}{|\Omega|} \sum_{(i,j)\in\Omega} \left(I(\tilde{x}_0)_{i,j} - I(x)_{i,j}\right)^2 \tag{5}$$

where $\Omega = \{(i,j) \mid S_{i,j} = 1\}$.

This loss **preserves and modulates overlaps instead of eliminating them**: heads biased by the IoU channel route attention to overlapping pairs, while the objective explicitly pulls predicted pairwise overlaps toward ground-truth magnitudes.

The final training loss is defined as:

$$\mathcal{L}_{\text{fin}} = \omega_{\text{snr}} \cdot \mathcal{L}_{\text{simple}} + \omega_{\text{snr}}^{\text{rec}} \cdot \mathcal{L}_{\text{rec}} + \omega_t \cdot (\mathcal{L}_{\text{alg}} + \mathcal{L}_{\text{overlap}}) \tag{6}$$

where ω_{snr} and $\omega_{\text{snr}}^{\text{rec}}$ are Min-SNR schedule weights for the diffusion and reconstruction losses respectively, ω_t is time dependent weight.

4 Experiments

4.1 Setup, Datasets and Tasks

We adopt the LACE continuous-diffusion backbone (DDPM with $\tilde{\mathbf{x}}_0$ reparameterization and masked-input unification) and layer our head-aware $N \times N$ masked-attention and RA representation on top.

Optionally, reading order (RO) is estimated with an XY-cut–style algorithm (Meunier, 2005) and encoded as a binary adjacency matrix (1 for allowed/adjacent links, 0 otherwise).

We consider two regimes: max 25 page items (standard) and max 50 page items (extended). We evaluate on the RICO dataset (Deka et al., 2017), which contains approximately 72k user interface designs from about 9.7k Android apps spanning 27 classes, and on the PubLayNet (Zhong et al., 2019) dataset which consists of 330k articles from PubMed Central™. We evaluate our model on three generation tasks defined in prior works (Rahman et al., 2021): conditional generation based solely on class (C→S+P), conditional generation based on both class and size (C+S→P), and completion in which attributes are given for a subset of elements and the remainder must be predicted. In our setting these tasks are *relation-conditioned* and correspond to the **layout-variation** task (Table 1): at both training and inference, LayoutLattice is additionally conditioned on the pairwise IoU (and, optionally, reading order) *computed from the ground-truth boxes of the layout*—the source layout whose relational structure is to be preserved. We make this explicit: these priors are the same ground-truth relations used for supervision, supplied to the model as a conditioning signal, exactly as a source layout is available in real layout-variation, refinement, and template-guided design. Only *relative* structure (pairwise overlap/order) is provided; absolute size and position are still generated, so the condition does not determine the target. Standard baselines are evaluated in their original setting without these pairwise inputs (see the flatten control, §4.2). All experiments are run on a single NVIDIA A10 GPU (24 GB VRAM), with Adam optimizer, batch size of 256 and 1000 diffusion steps. Learning rate starts at 1e-4 and decays with ReduceLROnPlateau schedule. Each model is trained for approximately 200 epochs at 5 minutes per epoch, for a total training time of 16.7 hours.

4.2 Quantitative Evaluation

We evaluate LayoutLattice on three benchmarks across the three conditional generation tasks. FID (Heusel et al., 2017) and Max. IoU are the primary metrics; lower FID and higher Max. IoU, which measures conditional similarity between generated and real layouts, indicates better realism and geometric fidelity, respectively. Reported numbers are single-run; across **20 seeds** FID varies within 2–5% and **Max. IoU within 3–5% relative**, and every claimed improvement exceeds this spread (Appendix O).

Table 2: Results on RICO, max 25 page items. Columns give the standard task; rows in the lower two blocks additionally receive the reference pairwise condition $\mathcal{R}$ (i.e., C+IoU(+RO)→S+P, etc.; Table 1), either flattened into per-token inputs (control) or as attention biases (ours). †quoted under the identical protocol; all other rows self-measured with the same extractor (Appendix M). ±: relative spread across 20 seeds (Appendix O).

Task	C → S+P		C+S → P		Completion	
Setting	FID ↓ (±2–5%)	Max. IoU ↑ (±3–5%)	FID ↓ (±2–5%)	Max. IoU ↑ (±3–5%)	FID ↓ (±2–5%)	Max. IoU ↑ (±3–5%)
No pairwise condition (original setting):						
LayoutTrans†	5.57	0.223	3.73	0.323	3.71	0.537
BLT†	17.4	0.202	4.48	0.340	117	0.471
BART†	3.97	0.253	3.18	0.334	8.87	0.527
MaskGIT†	26.1	0.262	8.05	0.320	33.5	0.533
VQDiffusion†	4.34	0.252	3.21	0.331	11.0	0.541
LayoutDM†	3.55	0.277	2.22	0.392	9.00	0.576
LACE†	3.24	0.340	2.87	0.418	4.45	0.527
Relation-conditioned, same $\mathcal{R}$ as flattened per-token input features (control):						
LACE + IoU (flatten)	19.35	0.316	20.14	0.375	17.39	0.325
LACE + IoU + RO (flatten)	18.83	0.312	14.81	0.320	12.66	0.324
Relation-conditioned, $\mathcal{R}$ as attention biases (ours):						
LayoutLattice (IoU)	2.72	0.430	2.03	0.459	3.71	0.454
LayoutLattice (RA + IoU)	2.12	0.546	1.85	0.553	3.28	0.537
LayoutLattice (RA + IoU + RO)	**1.52**	**0.574**	**0.94**	**0.593**	**1.91**	**0.579**

Table 3: Results on RICO, max 50 page items (all rows self-measured). Rows below the rule are relation-conditioned (Table 1). ±: relative spread across 20 seeds (Appendix O).

Setting	Task	C → S+P		C+S → P		Completion	
		FID ↓ (±2–5%)	Max. IoU ↑ (±3–5%)	FID ↓ (±2–5%)	Max. IoU ↑ (±3–5%)	FID ↓ (±2–5%)	Max. IoU ↑ (±3–5%)
No pairwise condition:							
LACE		11.62	0.085	9.46	0.191	16.3	0.073
Relation-conditioned, $\mathcal{R}$ as attention biases (ours):							
LayoutLattice (IoU)		2.73	0.420	2.02	0.460	3.98	0.441
LayoutLattice (RA + IoU)		2.31	0.528	1.76	0.538	3.53	0.537
LayoutLattice (RA + IoU + RO)		**1.45**	**0.561**	**0.87**	**0.581**	**1.89**	**0.568**

For the standard 25-item regime *(Table 2)*, we see a common trend across the three tasks: Attention bias plus RA reduce FID and increase Max. IoU consistently, indicating large gains in both realism and geometric structure. For the extended 50-item regime *(Table 3)* — where no prior art reports results — we trained the FID feature model and baseline from scratch (not re-tuned; trained to convergence; Appendix M). We see a big drop in scores for vanilla LACE model, and this is probably because of the impact of overlaps, as it plays a critical role in maintaining the quality of generated layouts as element count increases. As layouts become more crowded with more elements, preserving spatial coherence becomes significantly harder. The IoU bias directly addresses this by explicitly encoding overlap relationships between elements, and as a result, we observe strong improvements: FID drops from **11.62** → **2.73** on C→S+P and Max. IoU improves from **0.085** → **0.420**, and a similar improvement was observed for other tasks as well. Adding RA with page tokens further improves both scores. Predictions remain stable even with 50 page items on a dataset with large overlaps, highlighting the scalability of the approach.

Why we compare against the unconditioned baseline (flatten control). A natural question is whether the gains simply reflect access to the pairwise condition. The only way to give a standard backbone the *same* condition without changing its architecture is to flatten the $N \times N$ matrix into per-token input features. Table 2 ("flatten" rows) shows this *degrades* the backbone far below its no-input setting (e.g., C→S+P FID $3.24 \rightarrow 19.35$; the same holds for IoU+RO and on both backbones): the values are present but a per-token encoder has no way to bind entry $[i, j]$ to the correct element pair, so the optimizer chases a signal it cannot dereference. Flatten is **not under-tuned** (same backbone/budget/schedule/lr; the IoU features are supplied clean, never noised). Merely *having* the condition is thus insufficient—it must be injected through a channel that preserves pairwise structure, which is exactly our attention bias. Because injecting the priors as inputs hurts the baseline, we compare against the baseline in its strongest (unconditioned) configuration. Nor is the gain the metric echoing the condition: Max. IoU matches *absolute* boxes while the condition is *relative*, sparse overlap—flatten even scores *below* no-condition on Max. IoU (0.316 vs. 0.340).

Beyond the IoU channel, we also enabled reading-order (RO). Adding these contextual priors further reduces FID and increases Max. IoU, illustrating the flexibility of our interface: $N \times N$ contextual features can be injected as additive attention biases to dial in desired structure (e.g., RO, groups) without modifying the backbone.

Max. IoU on the completion task is a noisier signal than on other tasks; we therefore prioritize FID for completion (Appendix J).

On PubLayNet *(Table 4)*, a distinct domain of scientific document layouts, we consistently outperform prior arts across all tasks, confirming that the gains are not dataset-specific. To further verify that gains from attention bias are generalizable to other architectures, we replaced the backbone with LayoutDM and applied IoU and reading-order biases; the contextual attention interface consistently improves LayoutDM as well, yielding similar improvements (Appendix G, Table 5).

Table 4: Results on PubLayNet. Rows below the second rule are relation-conditioned (Table 1). †quoted under the identical protocol; all other rows self-measured with the same extractor (Appendix M). ±: relative spread across 20 seeds (Appendix O).

Setting \ Task	C → S+P		C+S → P		Completion	
	FID ↓ (±2–5%)	Max. IoU ↑ (±3–5%)	FID ↓ (±2–5%)	Max. IoU ↑ (±3–5%)	FID ↓ (±2–5%)	Max. IoU ↑ (±3–5%)
No pairwise condition (original setting):						
BART†	9.36	0.320	5.88	0.375	9.58	0.446
MaskGIT†	17.2	0.319	5.86	0.380	19.7	0.484
VQDiffusion†	10.3	0.319	7.13	0.374	11.1	0.373
LayoutDM†	7.95	0.310	4.25	0.381	7.65	0.377
LACE†	5.14	0.383	3.07	0.463	6.03	0.396
Relation-conditioned, $\mathcal{R}$ as attention biases (ours):						
LayoutLattice (IoU, 25)	2.90	0.517	2.36	0.528	4.36	0.520
LayoutLattice (IoU, 50)	2.85	0.516	2.80	0.527	4.34	0.521
LayoutLattice (RA + IoU, 25)	2.49	0.560	2.42	0.555	3.55	0.560
LayoutLattice (RA + IoU, 50)	2.53	0.556	2.45	0.558	3.55	0.557
LayoutLattice (RA + IoU + RO, 25)	**2.38**	**0.563**	**1.87**	**0.569**	**2.21**	**0.567**
LayoutLattice (RA + IoU + RO, 50)	**2.42**	**0.564**	**1.86**	**0.568**	**2.24**	**0.566**

4.3 Qualitative Evaluation

We show side-by-side samples against LayoutDM, LACE, and LayoutLattice models in both the 25 and 50-item regimes (Appendix H, Figure 3) on Rico dataset to visualize structural fidelity and overlap preservation.

Page-size variation: We also showcase qualitative examples of page-size variation across aspect ratios; see Appendix I (Figure 4) for the full visual results.

Attention head visualization: We visualize the attention weights of IoU-biased and free heads in Appendix A (Figure 2).

5 Conclusion and Future Work

Head-aware $N \times N$ contextual attention masks provide a simple, drop-in interface for injecting pairwise priors into transformers. By biasing a subset of heads with IoU/reading-order and supervising an overlap-consistency loss, the model learns graded overlaps instead of erasing them. Together with a RA encoding and a page token, our approach delivers large gains in FID and Max. IoU across 25- and 50-item regimes without architectural changes and with negligible additional overhead. We further demonstrate the generality of the mechanism on human pose prediction, where injecting skeleton priors as attention biases outperforms the state-of-the-art baseline (Appendix Q). **We recommend the $N \times N$ mask interface as a general pattern for injecting structural priors into attention.**

A current limitation is that some domain expertise is required to craft good priors, which may otherwise hurt performance. The method also assumes the pairwise relations are available as a condition, computed from a reference/source layout (as in layout variation/refinement); where no reference exists they must be estimated, and quality then depends on that estimate (further limitations: Appendix P). Promising directions for future work include extending the mask design across pages and sections to capture document-level structure, and exploring learned soft priors as content signals to complement hand-crafted geometry.

AI Use Statement

Generative AI tools were used only for writing and editing assistance and did not contribute to the core methodology, experimental design, or analysis. All AI-assisted text was reviewed for accuracy. We take responsibility for the final content of this work.

Ethics Statement

This work involves no human subjects, no scraped personal data, and no high-risk applications. Both RICO and PubLayNet are publicly available research benchmarks. The proposed model generates abstract layout bounding boxes and classes; it does not generate visual content and poses no known risk for misuse.

Reproducibility Statement

Implementation details needed to reproduce our results are provided throughout the paper: the model architecture and contextual attention mechanism are described in §3.3, and the training/evaluation setup (datasets, splits, optimizer, hyperparameters, compute) is given in §4.1. The evaluation protocol, including FID feature-extractor training and baseline provenance, is detailed in Appendix M, and run-to-run variability across 20 seeds is reported in Appendix O. Code is attached with the submission, and both datasets used (RICO, PubLayNet) are publicly available.

References

Diego Martin Arroyo, Janis Postels, and Federico Tombari. Variational transformer networks for layout generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13642–13652, 2021.

Jian Chen, Ruiyi Zhang, Yufan Zhou, Rajiv Jain, Zhiqiang Xu, Ryan Rossi, and Changyou Chen. Towards aligned layout generation via diffusion model with aesthetic constraints. In *International Conference on Learning Representations*, 2024.

Chin-Yi Cheng, Forrest Huang, Gang Li, and Yang Li. PLay: Parametrically conditioned layout generation using latent diffusion. In *International Conference on Machine Learning*, pp. 5449–5463, 2024.

Biplab Deka, Zifeng Huang, Chad Franzen, Joshua Hibschman, Daniel Afergan, Yang Li, Jeffrey Nichols, and Ranjitha Kumar. Rico: A mobile app dataset for building data-driven design applications. In *Proceedings of the 30th annual ACM symposium on user interface software and technology*, pp. 845–854, 2017.

Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, volume 27, 2014.

Kamal Gupta, Justin Lazarow, Alessandro Achille, Larry S Davis, Vijay Mahadevan, and Abhinav Shrivastava. LayoutTransformer: Layout generation and completion with self-attention. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1004–1014, 2021.

Tiankai Hang, Shuyang Gu, Chen Li, Jianmin Bao, Dong Chen, Han Hu, Xin Geng, and Baining Guo. Efficient diffusion training via min-snr weighting strategy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7441–7451, 2023.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851, 2020.

Mude Hui, Zhizheng Zhang, Xiaoyi Zhang, Wenxuan Xie, Yuwang Wang, and Yan Lu. Unifying layout generation with a decoupled diffusion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1942–1951, 2023.

Md Shamim Hussain, Mohammed J Zaki, and Dharmashankar Subramanian. Global self-attention as a replacement for graph convolution. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 655–665, 2022.

Naoto Inoue, Kotaro Kikuchi, Edgar Simo-Serra, Mayu Otani, and Kota Yamaguchi. LayoutDM: Discrete diffusion model for controllable layout generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10167–10176, 2023.

Zhaoyun Jiang, Jiaqi Guo, Shizhao Sun, Huayu Deng, Zhongkai Wu, Vuksan Mijovic, Zijiang James Yang, Jian-Guang Lou, and Dongmei Zhang. LayoutFormer++: Conditional graphic layout generation via constraint serialization and decoding space restriction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18403–18412, 2023.

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.

Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. In *International Conference on Learning Representations*, 2014.

Xiang Kong, Lu Jiang, Huiwen Chang, Han Zhang, Yuan Hao, Haifeng Gong, and Irfan Essa. BLT: Bidirectional layout transformer for controllable layout generation. In *European Conference on Computer Vision*, pp. 474–490, 2022.

Jean-Luc Meunier. Optimized xy-cut for determining a page reading order. In *International Conference on Document Analysis and Recognition*, pp. 347–351, 2005.

Ofir Press, Noah A Smith, and Mike Lewis. Train short, test long: Attention with linear biases enables input length extrapolation. In *International Conference on Learning Representations*, 2022.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

Soliha Rahman, Vinoth Pandian Sermuga Pandian, and Matthias Jarke. RUITE: Refining UI layout aesthetics using transformer encoder. In *Companion Proceedings of the 26th International Conference on Intelligent User Interfaces*, pp. 9–11, 2021.

Jaejung Seol, Seojun Kim, and Jaejun Yoo. Posterllama: Bridging design ability of language model to content-aware layout generation. In *European Conference on Computer Vision*, pp. 451–468, 2024.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 464–468, 2018.

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

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

Timo Von Marcard, Roberto Henschel, Michael Black, Bodo Rosenhahn, and Gerard Pons-Moll. Recovering accurate 3d human pose in the wild using imus and a moving camera. In *European Conference on Computer Vision*, 2018.

Chenxin Xu, Robby T Tan, Yuhong Tan, Siheng Chen, Xinchao Wang, and Yanfeng Wang. Auxiliary tasks benefit 3d skeleton-based human motion prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9509–9520, 2023.

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

Guangcong Zheng, Xianpan Zhou, Xuewei Li, Zhongang Qi, Ying Shan, and Xi Li. LayoutDiffusion: Controllable diffusion model for layout-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22490–22499, 2023.

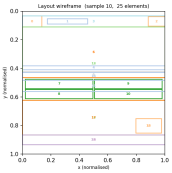
Xinru Zheng, Xiaotian Qiao, Ying Cao, and Rynson W H Lau. Content-aware generative modeling of graphic design layouts. *ACM Transactions on Graphics*, 38(4):1–15, 2019.

Xu Zhong, Jianbin Tang, and Antonio Jimeno Yepes. PubLayNet: Largest dataset ever for document layout analysis. In *International Conference on Document Analysis and Recognition*, pp. 1015–1022, 2019.

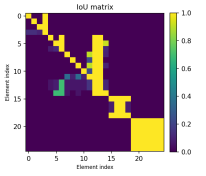
Appendix

A Attention Head Visualization

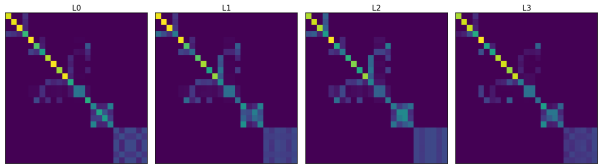
To validate that the contextual attention mechanism operates as designed, we visualize the average attention weights of IoU-biased heads and free heads across transformer layers (L0–L3) for a sample layout (Figure 2). The wireframe and its ground-truth IoU matrix are shown for reference. Bright line along diagonal is because IoU of item with itself is 1.



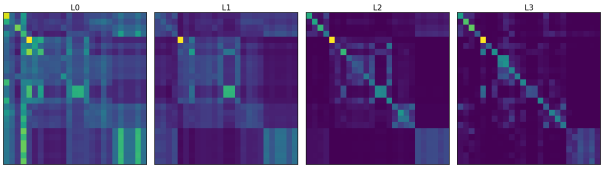
(a) Layout wireframe



(b) Ground-truth IoU matrix



(c) IoU heads (L0–L3)



(d) Free heads (L0–L3)

Figure 2: Attention head visualization for a 25-element RICO layout.

The *IoU heads* exhibit sparse, structured attention that closely mirrors the ground-truth IoU matrix: they attend strongly to element pairs with IoU > 0 and block attention to non-overlapping tokens. This sparsity pattern is stable and consistent across all four layers, demonstrating that the bias reliably enforces the intended relational structure throughout the network.

An interesting observation emerges in the *free heads*: at L0, they attend broadly across all tokens, as expected for unconstrained heads. By L3, however, the pattern becomes noticeably more structured, with higher weights appearing on many of the same pairs that carry non-zero IoU, despite no explicit overlap bias. This suggests that free heads, while unconstrained, do pick up on the geometric structure of the layout over depth, guided by the shared overlap training signal from IoU heads.

B Continuous Diffusion Preliminaries

We follow the DDPM formulation (Ho et al., 2020). A clean layout state $\mathbf{x}_0$ is gradually corrupted to $\mathbf{x}_t$ with a noise schedule, and a network learns to reverse the process by predicting noise and obtaining an estimate of the clean state $\tilde{\mathbf{x}}_0$. Layouts are represented as sequences of length L with a

padding class ($c = N$) and zero bounding boxes for absent items; all losses and masks operate only on valid tokens.

Forward process. A continuous-valued layout $\mathbf{x}_0$ is corrupted to noisy latents $\mathbf{x}_{1:T}$, converging to $\mathcal{N}(0, \mathbf{I})$ after T steps. The transition is $q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I})$ with variance schedule $\beta_1, \ldots, \beta_T$. The marginal at arbitrary t is:

$$q(\mathbf{x}_t \mid \mathbf{x}_0) = \mathcal{N}\big(\mathbf{x}_t;\ \sqrt{\bar{\alpha}_t}\,\mathbf{x}_0,\ (1-\bar{\alpha}_t)\mathbf{I}\big)\,, \tag{7}$$

where $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$ and $\alpha_t = 1 - \beta_t$.

Reverse process. The data $\mathbf{x}_0$ is reconstructed by sampling from $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ starting from $\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$. Using $\mathbf{x}_0$-parameterization, the mean estimate is:

$$\boldsymbol{\mu}_\theta(\mathbf{x}_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\,\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right), \tag{8}$$

where $\boldsymbol{\epsilon}_\theta$ is trained with the simplified objective:

$$\mathcal{L}_{\text{simple}} = \left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta\big(\sqrt{\bar{\alpha}_t}\,\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon},\ t\big)\right\|^2, \tag{9}$$

with $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$. The clean-sample estimate used for geometric losses is:

$$\tilde{\mathbf{x}}_0(\mathbf{x}_t) = \frac{1}{\sqrt{\bar{\alpha}_t}}\left(\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\cdot\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right). \tag{10}$$

C Multi-Head Attention and Masking Details

Transformers aggregate context via multi-head attention. Given an input sequence $\mathbf{X} \in \mathbb{R}^{N\times d}$ of N tokens with embedding dimension d, each of H attention heads computes queries, keys, and values through learned linear projections:

$$\mathbf{Q}_h = \mathbf{X}\mathbf{W}_h^Q, \quad \mathbf{K}_h = \mathbf{X}\mathbf{W}_h^K, \quad \mathbf{V}_h = \mathbf{X}\mathbf{W}_h^V, \quad (h = 1, \ldots, H) \tag{11}$$

where $\mathbf{W}_h^Q, \mathbf{W}_h^K, \mathbf{W}_h^V \in \mathbb{R}^{d\times d_k}$ are projection matrices and $d_k = d/H$ is the per-head dimension. Attention scores are computed as scaled dot-products between Q and K, with an additive bias $\mathbf{B}_h$:

$$\mathbf{S}_h = \frac{\mathbf{Q}_h\mathbf{K}_h^\top}{\sqrt{d_k}} + \mathbf{B}_h. \tag{12}$$

Attention weights are obtained via softmax normalization:

$$\mathbf{A}_h = \mathrm{softmax}(\mathbf{S}_h), \quad \mathbf{O}_h = \mathbf{A}_h\mathbf{V}_h. \tag{13}$$

Finally, the per-head outputs are concatenated and projected to produce the final output:

$$\mathbf{O} = \mathrm{concat}(\mathbf{O}_1, \ldots, \mathbf{O}_H)\mathbf{W}^O, \tag{14}$$

where $\mathbf{W}^O \in \mathbb{R}^{d\times d}$ is a learned projection that mixes the concatenated head outputs back into a single d-dimensional representation.

Effect of the bias under softmax. With the bias tensor $\hat{M}$ of Section 3.3, the attention weight for query i and key j is

$$A_h[i,j] = \frac{\exp\big(S_h[i,j]\big)}{\sum_{t=1}^{N}\exp\big(S_h[i,t]\big)} \tag{15}$$

(softmax over keys j for each query i)

From the softmax definition above, setting a biased logit to a large negative value (e.g., $S_h[i,j] \approx -10^4$, our numerical proxy for $-\infty$) yields $\exp(S_h[i,j]) \approx 0$. The denominator $\sum_t \exp(S_h[i,t])$ is then dominated by the remaining, non-masked keys, so the resulting attention weight $A_h[i,j]$ effectively becomes 0, blocking attention. In the Softmax function, as the logit moves far negative,

the probability collapses to zero; near zero the weight is moderate; for large positive logits, the weight dominates. Non-zero entries are preserved and act as non-negative additive biases (e.g., IoU scores or 1 from S).

Note: Step 4 mirrors the default behavior in common libraries (e.g., PyTorch's scaled_dot_product_attention): the bias matrix $\hat{M}$ from Step 3 is passed as the attention bias to multi-head attention (scaled dot-product attention). Boolean masks are internally converted to $-\infty$ on masked positions, while float masks are added directly to the logits—producing the same "blocked vs. biased" effect described above.

Head partitioning (masked vs free heads): we can dedicate only a subset of the H attention heads to receive $N \times N$ biases while leaving the remaining heads unmasked (bias $+1$ on valid pairs, -10^4 on padded; the constant $+1$ normalizes away under softmax). This preserves capacity for the model to discover new patterns as in the baseline, while ensuring that some heads always respect global context. Let $H_{\text{masked}} + H_{\text{free}} = H$. We broadcast M only to the first H_{masked} heads, and set the H_{free} heads to padding matrix.

D Relative-Area and Page-Token Details

From the representation (x, y, h, s), width is decoded as $w = s^2/h$ and clamped to the valid range. Normalization uses $\max(W, H)$ to match the page token.

Note: One could instead anchor on width and derive $h = a/w$; the choice is dataset/use-case dependent. For example, if auxiliary cues like font height are available, a height-anchored $(h, \sqrt{a})$ parameterization can be more natural.

Inference practice: we condition the page token (mask it as given) so predictions are made in the correct page context. For page-size variation, we swap in a new page token for the target aspect ratio and keep item relative-area attributes masked (preserved), letting the model adapt placements while maintaining size ratios.

E Alignment Loss

Following LACE (Chen et al., 2024), for each of the six alignments, coordinate difference matrix for the layout is defined as: $\mathbf{A}^*(\mathbf{x})_{i,j} = |b_i^* - b_j^*|, * \in \mathcal{A}$. Then the alignment loss function between a given prediction $\tilde{\mathbf{x}}$ at time t and a real layout $\mathbf{x}$ is defined as:

$$\mathcal{L}_{\text{alg}}(\tilde{\mathbf{x}}, \mathbf{x}) = -\log\left(1 - \frac{1}{6}\sum_{*\in\mathcal{A}} \frac{\left\|\mathbf{A}^*(\tilde{\mathbf{x}}) \circ \mathbf{1}_{\mathbf{A}^*(\mathbf{x})=0}\right\|_1}{\left\|\mathbf{1}_{\mathbf{A}^*(\mathbf{x})=0}\right\|_1}\right), \tag{16}$$

where $\mathbf{1}_{\mathbf{A}^*(\mathbf{x})=0}$ is the binary ground truth alignment mask matrix selecting the zero entries of the coordinate difference matrix of the true layout, which represents the true alignment pattern in $\mathbf{x}$, and $\circ$ is the Hadamard product.

F Timestep Weighting Rationale

We apply Min-SNR weighting (Hang et al., 2023) to both the diffusion loss and the reconstruction loss, for complementary reasons.

Diffusion (noise) loss. Predicting the added noise is hardest near early timesteps ($t \approx 0$), where the noise level is tiny and prediction errors are disproportionately large. Without weighting, the high SNR at small t causes these steps to dominate the training objective. Min-SNR clamps $\mathrm{SNR}(t)$ to γ, preventing these early steps from contributing unfair gradients.

Reconstruction ($\tilde{\mathbf{x}}_0$) loss. At early timesteps ($t \approx 0$), $\mathrm{SNR}(t)$ becomes very large. If the reconstruction loss were weighted in proportion to SNR, these trivially easy steps—where the layout is nearly clean and $\tilde{\mathbf{x}}_0 \approx \mathbf{x}_0$ almost for free—would dominate the objective and waste gradient budget on structure that requires no learning. Min-SNR again clamps $\mathrm{SNR}(t)$ to γ, down-weighting early high-SNR timesteps and directing training signal toward the harder, more noise-corrupted intermediate steps where precise geometric reasoning matters most.

G LayoutDM Backbone Ablation

To confirm that the improvements from the contextual attention interface are backbone-agnostic, we replace the LACE backbone with LayoutDM and apply the same IoU and reading-order (RO) bias channels. Results on the Rico dataset (Table 5) show consistent FID reduction and Max. IoU improvement across all tasks, mirroring the trends observed with the LACE backbone. The flatten control is likewise backbone-agnostic: injecting the same priors into LayoutDM as flattened per-token inputs degrades it far below the unconditioned baseline (Table 5), confirming that the gains come from the attention-bias interface, not from access to the priors.

Table 5: LayoutDM backbone ablation on RICO. The top row has no pairwise condition; all other rows are relation-conditioned (Table 1), either as flattened per-token inputs (control) or as IoU / reading-order attention biases. †quoted under the identical protocol; all other rows self-measured with the same extractor (Appendix M). ±: relative spread across 20 seeds (Appendix O).

Setting	Task	C → S+P		C+S → P		Completion	
		FID ↓ (±2–5%)	Max. IoU ↑ (±3–5%)	FID ↓ (±2–5%)	Max. IoU ↑ (±3–5%)	FID ↓ (±2–5%)	Max. IoU ↑ (±3–5%)
LayoutDM†		3.55	0.277	2.22	0.392	9.00	0.576
Same pairwise condition injected as flattened per-token input features (control):							
LayoutDM + IoU (flatten, 25)		21.32	0.201	21.10	0.287	18.44	0.237
LayoutDM + IoU + RO (flatten, 25)		18.89	0.229	16.46	0.302	16.72	0.297
LayoutDM (IoU, 25)		1.55	0.342	1.32	0.421	1.28	0.645
LayoutDM (IoU, 50)		1.62	0.334	1.27	0.428	1.37	0.632
LayoutDM (IoU + RO, 25)		0.89	0.378	0.69	0.458	0.81	0.674
LayoutDM (IoU + RO, 50)		**0.83**	**0.388**	**0.64**	**0.470**	**0.84**	**0.662**

H Qualitative Comparison

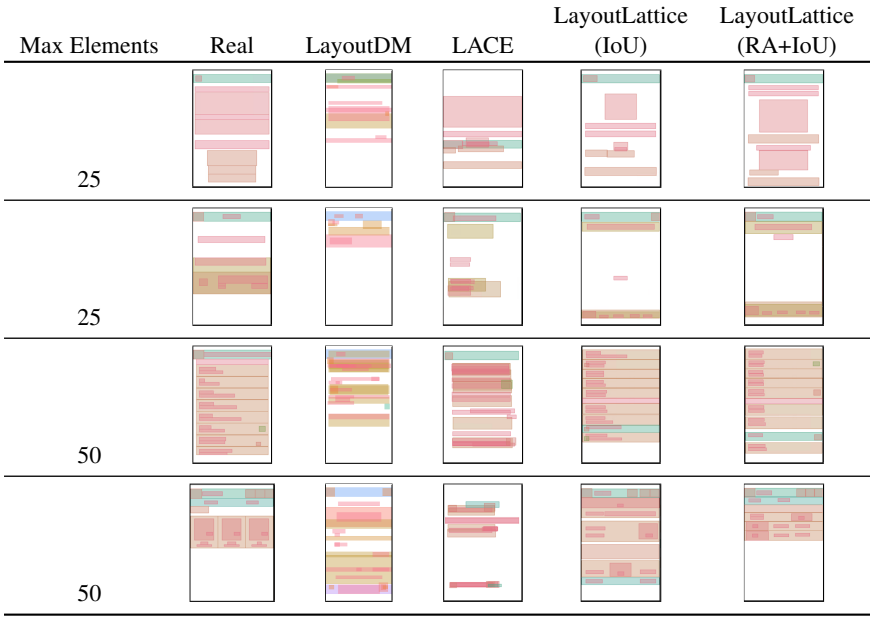


Figure 3: Comparison in conditional generation - C → S + P. LayoutDM and LACE are shown in their original (no pairwise condition) setting; LayoutLattice columns are relation-conditioned on $\mathcal{R}$ from the reference layout (Table 1).

I PAGE-SIZE VARIATION EXAMPLES

We showcase qualitative examples of resizing layouts across aspect ratios using our RA + IoU-biased model, demonstrating that these contextual masks enable controlled, structure-preserving page-size variation. As there is no established benchmark for this setting, we report visual evidence rather than standardized quantitative metrics.

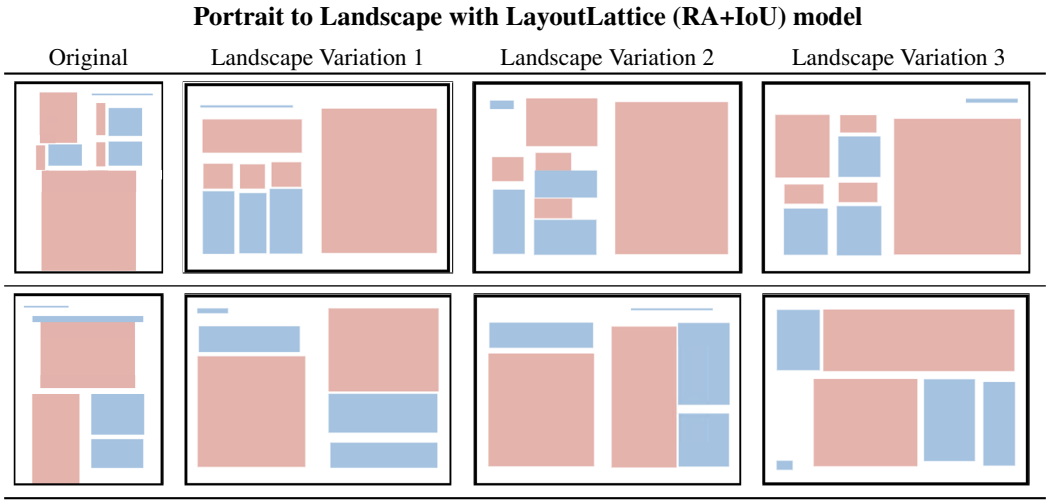


Figure 4: Page size variations with LayoutLattice (RA+IoU) model.

J NOTE ON MAX. IOU FOR COMPLETION

Max. IoU on the completion task is a noisier signal than on other tasks. As Inoue et al. (2023) observe, it becomes unstable when categories are also predicted — unlike C→S+P and C+S→P where categories are given. This is visible in the baselines: BLT achieves FID of 117 on completion yet still reports Max. IoU of 0.471. We therefore prioritize FID for completion; our Max. IoU gains there are consistent but should be interpreted alongside FID.

K NOTE ON READING ORDER

While XY-cut provides a simple and practical reading order estimate, we acknowledge it may not fully capture the non-linear structure of complex UI layouts. Nevertheless, when used as an attention bias channel alongside the IoU matrix, it proves highly effective: RA+IoU+RO consistently outperforms RA+IoU across all tasks and evaluation regimes. For example, on the RICO 25-item C+S $\rightarrow$ P task, adding reading order reduces FID from 1.85 to 0.94 and raises Max. IoU from 0.553 to 0.593.

More broadly, the intent behind including reading order is not to advocate for a particular estimation method, but to demonstrate that our $N \times N$ attention bias framework is a general mechanism capable of incorporating diverse pairwise signals beyond IoU. Reading order serves as proof-of-concept channels; the consistent improvements they yield across tasks and backbone architectures validate the extensibility of the approach.

L NOTE ON PERFORMANCE

Implementation overhead is minimal: we reuse standard additive attention-bias APIs (e.g., PyTorch scaled dot-product attention), so the backbone stays unchanged and the extra cost is only assembling M.

Pairwise attention biases scale quadratically with the number of elements N, which is an expected consequence of operating on $N \times N$ matrices. The $N \times N$ computation is not additional—it already

exists in standard self-attention, whose score matrix $QK^\top$ is itself $N \times N$. Our mechanism *reuses* it: the prior is an additive bias on the logits before softmax, with **no new parameters** and no change to the asymptotic cost ($O(N^2)$, already attention's cost). The only extra state is the prior matrix itself (a few $N \times N$ channels on the masked heads), negligible relative to the attention maps already present. At very large N one may slightly reduce the *training* batch size; **inference is unaffected**. Note that parameters do **not** scale with N (the N^2 is compute/memory, not weights)—this zero-parameter reuse is also what distinguishes our interface from heavier relational modules that must add trainable machinery to consume pairwise features. In practice, this overhead is modest and manageable:

- **Preprocessing cost.** Reading-order and group matrix construction is fully vectorized using standard Python matrix operations. For the full RICO dataset ($\sim$70k samples), this one-time preprocessing step completes in approximately 20 minutes.
- **Training cost.** At $N = 50$, one training epoch takes under 5 minutes, confirming that the quadratic overhead remains tractable at practical layout sizes.
- **Inference-time generalization.** A model trained at $N = 50$ generalizes to longer sequences at inference without retraining. We have successfully evaluated our 50-element model on pages with up to 100 elements. Inference can be extended to $N = 100$–150, covering the vast majority of real-world layout tasks and substantially exceeding the operating range of current state-of-the-art methods.
- **On-the-fly matrix computation.** For production deployment, reading-order and group matrices can be computed on-the-fly using the same vectorized operations, adding negligible latency to the inference pipeline.

M Evaluation Protocol and Provenance

FID protocol. FIDNetV3, trained per-dataset on the **train** split (identical to the LACE/LayoutDM protocol); FID over the **full test split—4,218 (RICO) / 11,142 (PubLayNet)**, one generated layout per test instance, same splits as prior work.

Baseline tuning. The 50-item LACE/LayoutDM baselines were **not re-tuned**—backbone/budget/schedule fixed by design so the delta reflects the added mechanism; they were trained to convergence.

Provenance. LayoutTrans/BLT/BART/MaskGIT/VQDiffusion/LayoutDM and 25-item LACE are **quoted** under the identical protocol (marked † in the tables); the 50-item rows, the flatten rows, and all LayoutLattice rows are **self-measured** with the same extractor.

Why LACE collapses at 50 items. Not under-tuning (trained to convergence). The cause is mechanistic: LACE's objective **explicitly penalizes overlap**, which conflicts with the abundant, intentional overlaps of crowded 50-item pages—precisely the failure mode we target. (FID is also unbounded/sensitive: poor models reach the *hundreds* on this benchmark—BLT 117 on completion (Table 2), our loss-only control 121 (Appendix N)—so LACE-50 at $\sim$11.6 is a *converged* model that is outperformed, not a broken run.)

N Disentangling Bias, Loss, RA, and RO

We isolate each factor: the **interface** (bias vs. same-condition flatten, Table 2); the **overlap-consistency loss** (a loss-only control *without* the bias fails to converge—FID 121.05 vs. 4.45, training losses $\sim 5\times$ higher); **RA** (IoU $\rightarrow$ RA+IoU); and **RO** (RA+IoU $\rightarrow$ RA+IoU+RO).

The bias *routes* (which pairs interact); the loss supervises *magnitude*. A **loss-only control (no bias) fails to converge**—FID 121.05 vs. 4.45 (backbone) on completion, training losses $\sim 5\times$ higher—because with no pathway to express pairwise IoU the loss *overwhelms* the diffusion/reconstruction/alignment objectives. RA and RO are isolated by the IoU $\rightarrow$ RA+IoU $\rightarrow$ RA+IoU+RO ladder. So the components are useful only together; the graded overlap is supervised by the loss (Figure 2 shows the routing).

O RUN-TO-RUN VARIABILITY

The numbers in the tables are from a single representative run. Since the model is inexpensive to train, we additionally re-trained it over **20 seeds** and find FID varies within **2–5%** and Max. IoU within **3–5% relative**, so every reported improvement exceeds this spread by a wide margin (e.g., Max. IoU 0.085→0.420 in the max-50-item regime).

P LIMITATIONS

(i) *Inference-time availability & source of the priors*—the method assumes the pairwise relations are available as a condition, computed from a reference/source layout (as in layout variation/refinement); where no reference exists they must be estimated, and quality then depends on that estimate. (ii) *Quadratic cost*—the $N \times N$ already exists in attention, so the bias adds no asymptotic cost, only cheap one-time construction (Appendix L). (iii) *Reading-order heuristics*—XY-cut is imperfect (Appendix K); when RO is noisy its contribution is only as good as the heuristic, but the mechanism is unaffected (RO is one optional channel; the IoU channel and free heads still operate). (iv) *Overlap vs. quality*—we do not maximize overlap; the loss **preserves** the source's original overlaps (designer intent).

Q ATTENTION BIASES BEYOND LAYOUT: HUMAN POSE PREDICTION

To support the claim that pairwise attention biases are a domain-agnostic mechanism, we apply the same $N \times N$ bias interface to human pose prediction—a task structurally unrelated to layout generation—and show that anatomically-grounded priors improve a strong transformer baseline, adding zero trainable parameters.

Task. Human body pose prediction forecasts the 3D positions of all body joints over a future time horizon given a short observation window of past motion. Formally, given joint positions across T_{obs} frames, the model must predict the next T_{fut} frames, where each frame contains 3D coordinates for all 23 body joints. The skeleton imposes well-known coordination patterns across joint pairs—bilateral symmetry between corresponding left/right joints, contralateral coupling between opposing arm and leg joints during locomotion, and the spine's role as the central kinematic anchor. A transformer attending over joints has no *a priori* knowledge of these relationships and must recover them entirely from data, expending model capacity on structure that is already known. This inefficiency is most damaging at long prediction horizons, where training statistics become too sparse to reliably estimate cross-body coordination patterns.

Dataset and baseline. We evaluate on 3D Poses in the Wild (3DPW) (Von Marcard et al., 2018), an in-the-wild benchmark with IMU-based ground-truth annotations across 60 sequences of diverse outdoor activities. Performance is mean per-joint position error (MPJPE, mm) at six horizons: 100, 200, 400, 600, 800, and 1000 ms. Our baseline is AuxFormer (Xu et al., 2023), the prior state of the art on 3DPW, which interleaves temporal and spatial self-attention in a masked-autoencoder framework. Its spatial attention treats all joint pairs uniformly with no structural bias—exactly the gap our $N \times N$ bias interface addresses.

Pairwise skeleton priors. We construct seven scalar features for every ordered joint pair (i, j), computed solely from the observed trajectory without any learnable parameters. Following the same interface described in Section 3, each feature is injected as an additive bias into one spatial attention head before the softmax. Joint pairs with no structural relationship for a given prior receive a bias of -10^4, hard-masking them from that head's attention.

- **Velocity direction coherence.** Temporal consistency of velocity-vector alignment for joint pair (i, j); restricted to pairs within k hops on the skeleton graph.
- **Bilateral speed symmetry.** Cross-correlation of CoM-relative speeds for anatomically symmetric pairs (e.g. left/right knee, shoulder, ankle).
- **Contralateral speed coupling.** Cross-correlation of CoM-relative speeds between contralateral pairs (right arm / left leg and vice versa), encoding the anti-phase arm–leg rhythm of bipedal locomotion.

- **Bilateral position symmetry.** Position-domain counterpart of bilateral speed symmetry, capturing quasi-static body shape symmetry.
- **Contralateral position coupling.** Position-domain counterpart of contralateral speed coupling.
- **Spine–body coherence.** Measures how tightly each spine joint (neck, thorax, pelvis) tracks the whole-body CoM trajectory.
- **Bilateral joint-angle coherence.** Cross-correlation of joint angles for symmetric pairs, capturing coordinated flexion–extension patterns.

AuxFormer's spatial attention uses 8 heads. We assign one prior to each of the seven heads above, leaving the eighth head entirely unbiased—mirroring the free-head design of Section 3.3—so the model retains capacity to discover data-driven patterns not captured by the hand-designed priors. Prior injection is applied exclusively in the encoder; the decoder receives no structural bias, leaving it free to synthesise encoder features without structural interference.

Results. Table 6 reports MPJPE across baselines and our method; we achieve state of the art at every horizon. The gain distribution is consistent with the design intent of the seven priors: the largest *relative* improvements occur at short horizons (100–200 ms; 6.1% and 2.9%), where velocity-direction and bilateral priors are expected to sharpen local joint attention, and the largest *absolute* improvements occur at long horizons (600–800 ms; 2.56–2.72 mm), where contralateral coupling priors are expected to anchor whole-body locomotion rhythms that sparse long-horizon training statistics cannot reliably recover.

These results echo the conclusions from layout generation: non-learnable pairwise priors and data-driven attention are complementary. The priors encode what is structurally invariant across all sequences; attention learns what varies with activity, speed, and individual style. Together, they validate the $N \times N$ bias interface as a general pattern for injecting structural priors into transformers, independent of domain.

Table 6: Average MPJPE (mm) on 3DPW. **Bold**/underline denote best/second-best.

Method	100 ms	200 ms	400 ms	600 ms	800 ms	1000 ms
Res-sup.	102.28	113.24	173.94	191.47	201.39	210.58
CSM	57.83	71.53	124.01	155.16	174.87	187.06
Traj-GCN	16.28	35.62	67.46	90.36	106.79	117.84
DMGNN	17.80	37.11	70.38	94.12	109.67	123.93
HisRep	15.88	35.14	66.82	93.55	107.63	114.75
MSR-GCN	15.70	33.48	65.02	93.81	108.15	116.31
STSGCN	18.32	37.79	67.51	92.75	106.65	112.22
PGBIG	17.66	35.32	67.83	89.60	102.59	109.41
SPGSN	15.39	32.91	64.54	91.62	103.98	111.05
AuxFormer (Xu et al., 2023)	<u>14.21</u>	<u>30.04</u>	<u>58.50</u>	<u>89.45</u>	<u>100.78</u>	<u>107.45</u>
Ours (w/ Skeleton Priors)	**13.34**	**29.16**	**58.07**	**86.89**	**98.06**	**105.87**

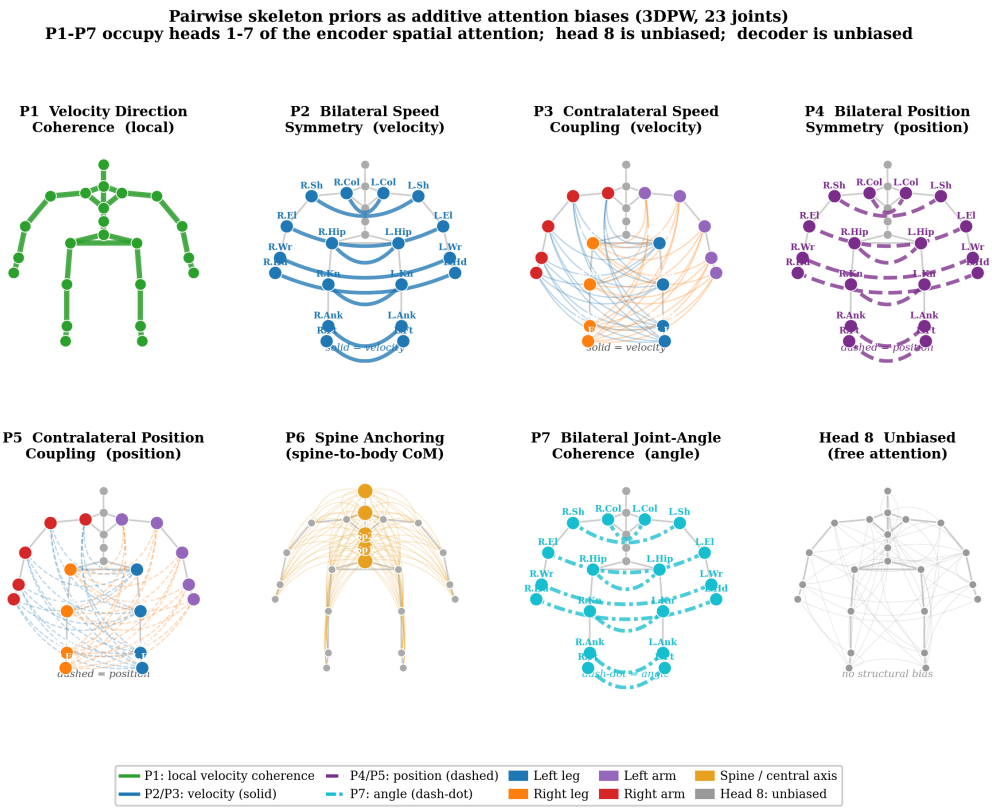


Figure 5: Visualization of the seven pairwise skeleton priors injected as attention biases. Each panel shows the $N \times N$ prior matrix for one head, where brighter entries indicate stronger structural coupling between the corresponding joint pair.