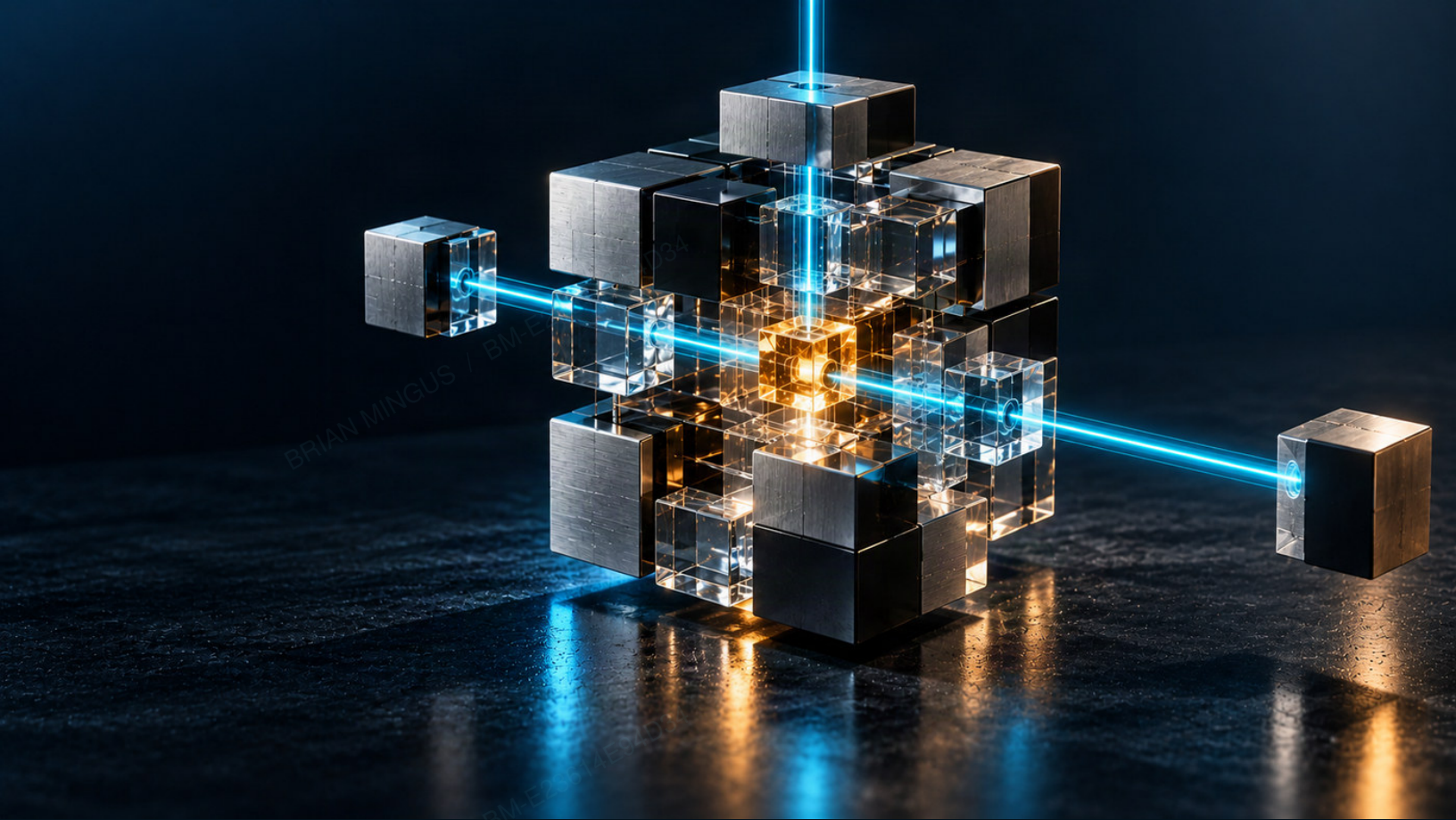




COMPUTATIONAL CREATIVITY | DEBATE SPARKS DRAFT

Three Coneheads Are All You Need

From Visual Allusion to an Executable Transformer

Brian Mingus

A 2006 cognitive science poster becomes a series of working Transformers, turning visual allusions into computational commitments and tests.

Added October 17, 2026. The complete four-page manuscript follows.

AI-generated conceptual artwork; separate from scientific figures.

brianmingus.com

Three Coneheads Are All You Need: From Visual Allusion to an Executable Transformer

Paper type: Debate Sparks

Brian Mingus

Abstract

Three Coneheads occupy one face of a cube. Read as a visual pun, they suggest multi-head attention. Take the rest of the picture seriously, and a collection of rotating, interlocking, partly blank cubes begins to suggest a computational architecture. Here, I turn that reading of a cognitive science club poster into a series of working Transformers. A spatial version implements three attention heads, axial rotary positions, local and global attention, reference cross-attention, and sparse expert routing. It learns two synthetic symbol tasks, with held-out accuracies of 66.75% and 100%. I read the recurring Homer/anatomical-brain pairing as an allusion to induction heads: recognize a recurring image and recall its companion. Assigning an order to those paired faces makes that association executable as predecessor matching and successor copying. Variable-gap probes verify the integrated heads' effect; the earlier sequence model's fixed-cycle success did not establish this behavior. The implementation makes the interpretation testable, while leaving open how much the picture actually constrained it. I propose that computational creativity should evaluate this transition from visual allusion to executable commitments as a creative act in its own right. The resulting debate concerns where the creativity occurs, how to distinguish a useful interpretation from a retrospective explanation, and what evidence would show that the image changed the design.

Take the Coneheads Seriously

I created the poster in 2006 for the University of Colorado Boulder Cognitive Science Club's "Brain Games!" event (Figure 1). It depicts a cube assembled from smaller cubes, populated by brains, skulls, a university mascot, fried eggs, and other images. Several detached cubes surround the assembly. On the cube at the back right are three Coneheads.

My reading is direct: the three Coneheads are a multi-head attention metaphor. For this project, I gave an AI coding assistant an equally direct instruction: implement the algorithm. This changed the work. It was no longer sufficient to describe a resemblance. An attention head had to become an actual learned operation; a rotation had to act on something; the spaces between cubes had to acquire computational consequences.

I take this as a useful provocation for computational creativity. An informal image can become a specification



Figure 1: The 2006 "Brain Games!" poster. Three Coneheads appear on the detached cube at the back right. Repeated symbols, spatial relations, blank faces, and detached cubes provide further material for interpretation.

through decisions that make its allusions executable. Those decisions can be recorded, challenged, and tested. The question is how much creative work happens in that transition, and how to recognize it.

This paper concerns the present act of turning those visual allusions into executable operations. I use published Transformer attention (Vaswani et al. 2017) and rotary position encoding (Su et al. 2021), and document the implementation choices needed to make the reading work.

From Allusion to Commitments

An interpretation becomes useful to an implementer when it excludes some choices. Taking the Coneheads literally commits me to three attention heads, each with learned query, key, and value projections. Taking the cubes' spatial arrangement seriously commits me to explicit coordinates. A program that only changes because its input array was re-ordered would miss this second commitment.

Table 1 records the main correspondences. They have different strengths. Three visible heads provide a particularly

Poster cue	Executable commitment
Three Coneheads	Three attention heads
Repeated image types	Eight learned symbol identities
Recurring Homer/brain pair	Induction: recognize an image, recall its paired companion
Cubes in three dimensions	Axial RoPE over x, y, z
Different faces	Optional face-orientation embeddings
Blank faces	Masked symbol reconstruction
Neighbors and an assembly	Local, then global attention
Detached cubes	Cross-attention to reference tokens
Different brain depictions	Sparse expert routing
Cube reorientation	Augmentation by 24 proper rotations

Table 1: A record of the adopted reading. Each right-hand entry is an implemented interpretation; the image does not determine its full mathematical definition.

compact pun. Blank faces suggest a missing-information problem, although a blank face does not establish that anything was hidden there. Different brain depictions suggest specialization, but the step from specialization to a mixture of experts is a larger interpretive leap. I retain that leap as a documented design choice.

The distinction matters most for rotation. RoPE rotates paired query and key coordinates according to position. Physically turning a pictured cube is a different operation. I connect them by assigning three groups of feature coordinates to the three spatial axes, following an axial approach developed for two-dimensional vision (Heo et al. 2024). For a token at $p = (x, y, z)$, the head uses

$$R(p) = \text{diag}(R_x(x), R_y(y), R_z(z)),$$

where each axis block contains two-dimensional rotations at fixed frequencies. Attention then computes

$$A_{ij} = \text{softmax}_j \left(\frac{(R(p_i)q_i)^\top R(p_j)k_j}{\sqrt{d_h}} + M_{ij} \right).$$

Here M_{ij} is zero for an allowed connection and negative infinity otherwise. Values are combined using A_{ij} . Relative displacement affects the scores; a common translation preserves the rotary dot products. This does not make the network exactly equivariant to physical cube rotations. Those rotations are supplied separately as training augmentation.

Other choices have no specific visual justification. The projection equations, frequency schedule, widths, depth, optimizer, and loss are implementation decisions. RMSNorm (Zhang and Sennrich 2019), SwiGLU (Shazeer 2020), and tied symbol embeddings complete a practical model. A useful specification should expose where its commitments end and the implementer’s decisions begin.

Implementation and Results

The first model is a small causal Transformer: two blocks, width 48, three heads, and one-dimensional RoPE. It has 57,024 parameters. Its task is to continue an eight-symbol

Task	Before	Trained	No geometry
Completion	11.37	66.75	31.29
Reference lookup	12.11	100.00	29.73

Table 2: Held-out symbol accuracy (%). Completion scores 2,304 masked symbols; lookup scores 6,912. “No geometry” disables both axial RoPE and the local mask at inference, with the trained weights fixed.

cycle after seeing it once. Training samples from a pool of 4,784 cycle classes; 256 classes are held out, including every cyclic rotation. After 300 updates, the model reaches 100% accuracy on 6,400 scored continuation tokens. It also generates the next 16 tokens exactly for 32 held-out prompts. This establishes a working sequence model, but it leaves most of the poster’s spatial structure unused.

The second model makes those relations explicit. It has two residual blocks of width 72, three heads of width 24, and 271,872 parameters. Each head allocates eight coordinates to each spatial axis. The first block restricts self-attention to Manhattan distance one; the second permits global attention. Both can attend to separately positioned reference tokens. Each block also routes a token to two of three SwiGLU experts, using learned gates and a load-balancing loss. This implements sparsely gated conditional expert computation (Shazeer et al. 2017); no semantic role is assigned to an expert in advance.

Inputs are manually named symbols and supplied coordinates. The model does not extract objects or geometry from poster pixels. The synthetic data represent a 3×3 symbol plane copied through three layers to form a cube. In *completion*, one copy in each column is masked, leaving nine symbols to predict. In *reference lookup*, all 27 cube symbols are masked and a separate nine-symbol plane supplies the reference information. At each scored cube location, the symbol is replaced by a mask token; matching symbols remain available elsewhere.

Training alternates these tasks for 600 updates, with batch size 32 and seed 2006. Each example jointly shuffles tokens and coordinates and samples one of the 24 proper cube rotations. Entire plane classes under the eight square rotations and reflections are held out. The run encounters 19,194 distinct training classes, with no overlap with the 256 held-out classes. Evaluation samples one seeded rotation and ordering per held-out pattern; it does not exhaust every orientation of every pattern.

Table 2 shows that the model learns these tasks and depends on the supplied geometry. Changing every reference symbol while retaining its position reduces agreement with the original lookup answers to 0%. This demonstrates dependence on reference contents; it may reflect correct adherence to the changed references. It is not a measure of robustness to incorrect memory.

These are inference controls, not independently retrained baselines. Disabling geometry removes two mechanisms together, so the control cannot isolate their contributions. The experiment has one training seed, and symbols within a pattern are correlated. Completion remains imperfect. Face

embeddings and causal mode in the second model are tested code paths, but its saved weights are trained on whole-cube bidirectional tasks. Sparse expert dispatch is implemented; the local attention mask still uses a dense score matrix. No efficiency gain is claimed.

The accompanying artifact contains source, checkpoints, results, and 28 passing numerical and behavioral tests across the two models. Tests include causal isolation, rotary identities, memory effects, expert dispatch, and data separation under symmetry. Reloaded checkpoints reproduce their saved reference logits exactly. These checks establish properties of the programs. They do not measure the creativity of the interpretation.

Induction Is More Than Repetition

The induction allusion begins with a specific pair. On the detached lower-left cube, Homer’s X-ray sits beside an orange anatomical brain. The upper-left cube in the main assembly repeats that pairing, with the two visible faces reversed. I read the recurrence as relational recall: recognize an image and recover its companion across changes of location and orientation. Let Homer be A and the anatomical brain be B . Assigning an order to the paired faces turns that association into $[A][B] \dots [A] \rightarrow [B]$, the induction pattern (Olsson et al. 2022). The picture supplies the recurring association; my traversal supplies its sequence order.

Each circuit operation retains that visual connection. The shared cube binds the two faces: a *previous-token head* carries A ’s identity to B ’s position. The recurring Homer image supplies the match: the later query–key (QK) circuit recognizes A in that carried identity. The companion face supplies what to recover: the value–output (OV) circuit copies B . Passing the paired identity into the later keys realizes *K-composition* (Elhage et al. 2021). These connected operations implement the adopted reading of the pair; the three Coneheads separately motivate the parallel head count.

The poster also gives Homer another companion: the CU logo on the upper-right cube. Under the same encoding, one image can therefore participate in competing associations. The induction head returns a mixture when several matching pairs have different successors, rather than assuming every image has one unique companion.

This reading also changes what I must test. A model can continue a fixed eight-symbol cycle by copying from a fixed offset. Induction should respond to the matching content when the gap varies. Changing the earlier B to C should change the predicted successor to C . Both checks are necessary before treating successful repetition as evidence for the proposed mechanism.

I test the unchanged sequence checkpoint on 256 paired prompts at each of 12 gap lengths from zero to 24. Pair members differ only at the earlier successor: $AB \dots A$ versus $AC \dots A$. Distractors exclude A , B , and C , ensuring a unique antecedent. The original cycle control remains at 100%, but the new probe yields 761 correct predictions out of 6,144, with both members correct in only one of 3,072 pairs (Table 3). These prompts differ from the training distribution; failure does not establish that all induction circuitry is absent.

Implementation	Prompt	Paired
Original learned model	12.39	0.03
Integrated induction heads	100.00	100.00
Same model, heads disabled	1.90	0.81

Table 3: Variable-gap accuracy (%). Prompt accuracy scores 6,144 continuations; paired accuracy requires both the original and changed successor to be correct in each of 3,072 pairs. The integrated heads and mixture gate are fixed constructions; these are behavior and ablation checks, not a learned-model benchmark.

I integrate this reading into the spatial Transformer as two fixed attention stages in its causal forward pass: previous-position attention, then one-hot content matching and successor-value copying. The match logit is 20. A causal mask prevents access to later symbols; MASK cannot serve as a query, antecedent, or successor. With a matching pair, the output mixes 95% induction probability with 5% base-model probability. Without a match it retains the base scores. This adds an explicit attention branch and output gate, rather than identifying a circuit learned by the original blocks.

The integrated model answers all 6,144 prompts correctly, including every changed successor. Disabling its induction heads reduces accuracy to 1.90%. This ablation tests the added branch’s effect; the base weights were trained for spatial reconstruction, and no additional optimization is performed. Bidirectional reconstruction retains exactly the V2 outputs, and saving and reloading preserves the causal scores. Tests also cover no-match fallback, ambiguity, masking, causality, and gradients.

Where Did the Creativity Occur?

The interpretive process has several candidates for creative credit: my original visual composition; my reading of the Coneheads as multi-head attention; the assistant’s selection of computational meanings; and the successive implementation and testing of those meanings. The trained miniature Transformer is a product of this process. It is not itself the agent that read the poster and proposed the architecture.

This distinction has precedent in computational creativity. The FACE and IDEA models describe creative acts, their framing, and the development and appreciation surrounding them (Colton, Charnley, and Pease 2011). In this case, the framing is unusually consequential: the request to treat an image as a specification turns an observation into a demand for construction. The architecture and the account of its origin become connected objects of evaluation. Jordanous’s evaluation methodology likewise asks that the relevant conception of creativity be made explicit before selecting tests (Jordanous 2012). Next-symbol accuracy answers a computational question, while leaving the creative question unresolved.

There is an obvious objection: a knowledgeable assistant could attach familiar Transformer features to almost any sufficiently complicated image. Given enough interpretive freedom, the cube might equally suggest a graph network, a cellular automaton, or a symbolic puzzle solver. An im-

plementation would then demonstrate programming competence without showing that the artwork contributed much. The present experiment does not rule this out.

That objection suggests a stronger experiment. Fix the reading before writing the code. Ask separate teams to produce mappings from the same poster, and record where they agree. Change one cue while leaving the rest intact: replace the three Coneheads with one head, remove the detached cubes, or erase the spatial arrangement. Test whether the proposed designs change in the predicted ways. If every version produces the same architecture, the image has done little constraining work.

Next, compare visual and textual starting points under the same time and compute budget. One group receives the poster, another an explicit feature list, and another an unrelated image of similar complexity. Evaluate executable correctness, the number of justified design commitments, and the diversity and usefulness of the resulting programs. Independent judges should see both the source and the mapping, and should distinguish features that constrain behavior from names attached after implementation. These comparisons are proposals; they have not been conducted here.

Such a study would also allow an unsuccessful interpretation to be informative. A metaphor that yields an elegant but untrainable program differs from one that yields a useful program through many unmotivated additions. The present completion result already resists a perfect story: the network gets roughly one third of the missing symbols wrong. Recording that result preserves the possibility that a compelling reading is a poor engineering choice.

The Spark

I propose treating visual interpretation as a source of provisional computational specifications. A commitment earns its place by connecting something visible to an operation with consequences. The interpretation then remains open to alternatives, counterfactual images, and empirical failure.

Three questions follow. How much must an image constrain a program before we credit the interpretation as design? Can a familiar architecture acquire creative value through a new route to its construction? And how should credit be distributed when a human insists on a reading and an AI supplies most of the executable detail?

The Coneheads provide a memorable starting point. The larger opportunity is to make the passage from metaphor to mechanism available for inspection. We should be able to turn the cube, change the heads, remove the reference, and see what the interpretation requires us to build differently.

Author Contributions and AI Assistance

I created the poster, proposed the multi-head attention interpretation, and directed the implementation and its extension. OpenAI Codex assisted with interpretation, generated and revised code, ran experiments and checks, and assisted with this manuscript. This account describes one human–AI development process; no controlled creativity study or independent human evaluation has been completed.

References

- Colton, S.; Charnley, J.; and Pease, A. 2011. Computational creativity theory: The FACE and IDEA descriptive models. In *Proceedings of the Second International Conference on Computational Creativity*, 90–95.
- Elhage, N.; Nanda, N.; Olsson, C.; Henighan, T.; Joseph, N.; Mann, B.; Askell, A.; Bai, Y.; Chen, A.; Conerly, T.; DasSarma, N.; Drain, D.; Ganguli, D.; Hatfield-Dodds, Z.; Hernandez, D.; Jones, A.; Kernion, J.; Lovitt, L.; Ndousse, K.; Amodei, D.; Brown, T.; Clark, J.; Kaplan, J.; McCandlish, S.; and Olah, C. 2021. A mathematical framework for transformer circuits. *Transformer Circuits Thread*.
- Heo, B.; Park, S.; Han, D.; and Yun, S. 2024. Rotary position embedding for vision transformer. In *European Conference on Computer Vision*.
- Jordanous, A. 2012. A standardised procedure for evaluating creative systems: Computational creativity evaluation based on what it is to be creative. *Cognitive Computation* 4(3):246–279.
- Olsson, C.; Elhage, N.; Nanda, N.; Joseph, N.; DasSarma, N.; Henighan, T.; Mann, B.; Askell, A.; Bai, Y.; Chen, A.; Conerly, T.; Drain, D.; Ganguli, D.; Hatfield-Dodds, Z.; Hernandez, D.; Johnston, S.; Jones, A.; Kernion, J.; Lovitt, L.; Ndousse, K.; Amodei, D.; Brown, T.; Clark, J.; Kaplan, J.; McCandlish, S.; and Olah, C. 2022. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*.
- Shazeer, N.; Mirhoseini, A.; Maziarz, K.; Davis, A.; Le, Q.; Hinton, G.; and Dean, J. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*.
- Shazeer, N. 2020. GLU variants improve transformer. *arXiv preprint arXiv:2002.05202*.
- Su, J.; Lu, Y.; Pan, S.; Murtadha, A.; Wen, B.; and Liu, Y. 2021. RoFormer: Enhanced transformer with rotary position embedding. *arXiv preprint arXiv:2104.09864*.
- Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30.
- Zhang, B., and Sennrich, R. 2019. Root mean square layer normalization. In *Advances in Neural Information Processing Systems*, volume 32.