

Appendix X. Minimal Holor Calculus for Attention-Based Models

In this appendix we sketch a minimal instantiation of holor calculus for attention-based models. The aim is to show how a few simple, computable quantities in a transformer can be interpreted as approximations to the central holor objects (Φ , T_χ , R_e) and constraints (IAR, HC8), and how they induce holor-regularized learning dynamics as in Holor Calculus III.

We emphasize that this construction is deliberately minimal and layer-local. It does not reproduce the full continuum theory of Holor Calculus I-II, but serves as an accessible bridge for practitioners.

X.1 Awareness Manifold and Discrete Holor Proxies

Consider a single attention layer of a transformer. Let the input sequence length be L , with token indices:

This serves as a discrete awareness manifold for the layer.

- For each ($i \in M$):
 - The holor fibre (E_i) is the representation space at that layer; e.g., ($E_i \cong \mathbb{R}^{d_{\text{model}}}$).
 - The token's representation ($x_i \in E_i$) plays the role of the base holor field value ($H(i)$).
- For attention head (h), attention weights are:

We interpret:

- ($A_{ij}^h \approx \Phi^h(i \rightarrow j)$): the discrete awareness flow field (Φ).

In this minimal construction we do not explicitly build:

- The trace space (T), nor the full holor connection.

Instead we define discrete quantities that:

- Approximate awareness scope and balance (IAR),
- Approximate torsion-memory effects (T_χ),

-
- Encode ethical admissibility (HC8).
-

X.2 IAR-style Regularization via Token-wise Entropy

Define the discrete awareness distribution for each head and token:

with entropy:

For bounds ($H_{\min} < H_{\max}$), define the IAR regularization loss:

This softly penalizes overly concentrated or overly diffuse awareness flows.

X.3 Loop Torsion: Short Cycles as T_{χ} Proxy

We treat ($A^{(h)}$) as a row-stochastic transition matrix. Powers capture multi-step flows:

Loopiness score at token (i):

Aggregate torsion loss:

This penalizes tokens that recurrently attend to themselves or local cliques.

X.4 Ethical Admissibility as Attention Constraint

Let ($E_{\text{mask}} \in \{0,1\}^L$) mark ethically problematic tokens.

Define the ethically problematic inflow:

With threshold ($\alpha \geq 0$), define:

This discourages awareness flows into ethically marked regions.

X.5 Combined Holor Energy and Loss

Single-layer holor loss:

Full loss over layers (with weights (β_{ℓ})):

Holor-regularized training objective:

This defines a projected descent:

Practitioners may use these components to experiment with holor-inspired constraints in transformer models.

PyTorch-style Implementation

```

import torch
import torch.nn.functional as F

def holor_regularizer(
    attn,
    eth_mask=None,
    H_min=0.5,
    H_max=2.5,
    w_IAR=1.0,
    w_loop=1.0,
    w_eth=1.0,
    alpha=0.0,
    eps=1e-12,
):
    B, H, L, _ = attn.shape

    # Entropy-based IAR term
    ent = -(attn * (attn + eps).log()).sum(dim=-1)
    low_v = torch.clamp(H_min - ent, min=0.0)
    high_v = torch.clamp(ent - H_max, min=0.0)
    L_IAR = (low_v**2 + high_v**2).mean()

    # Loop torsion via A^2 and A^3
    attn_sq = torch.matmul(attn, attn)
    attn_cu = torch.matmul(attn_sq, attn)
    loop2 = attn_sq.diagonal(dim1=-2, dim2=-1)
    loop3 = attn_cu.diagonal(dim1=-2, dim2=-1)
    loop_score = loop2 + loop3
    L_loop = loop_score.mean()

    # Ethical flow constraint
    if eth_mask is not None:
        mask = eth_mask[:, None, None, :].float()
        E_flow = (attn * mask).sum(dim=-1)
        ethViolation = torch.clamp(E_flow - alpha, min=0.0)
        L_eth = (ethViolation**2).mean()
    else:
        L_eth = torch.tensor(0.0, device=attn.device, dtype=attn.dtype)

    return w_IAR * L_IAR + w_loop * L_loop + w_eth * L_eth

```