Recursive Coherence Language Model (RCLM): Architecture & Pipeline

Nicolas Calder n926fl@gmail.com

October 24, 2025

Abstract

The architecture and pipeline of a Recursive Coherence Model, is built atop a standard Transformer engine and augmented with a recursive coherence controller. The controller operates on a density-like state derived from objective and subjective projections of hidden representations, explicitly optimizing entropy reduction and coherence amplification under task constraints. We outline modules, operators, training objectives, inference loop, and scaling considerations, and provide a practical implementation sketch in

Introduction

A recursive coherence model, using a quantum physics based mathematical framework of (informational/shannon and quantum informational / von neuman + thermodynamic) entropy reduction through recursive quantum coherence

Using the structure as

```
language >
objectivity and subjectivity >>
definitiveness and ambiguity >>>
coherence and entropy >>>>
```

Contents

$$\psi^{(O)} = P_O\, h, \qquad \psi^{(S)} = P_S\, h \tag{1}$$

(per token or per segment).
Definitiveness / Ambiguity → mixed state:

$$\rho_0 = \alpha\, \psi^{(O)} {\psi^{(O)}}^\top + (1-\alpha)\, \psi^{(S)} {\psi^{(S)}}^\top, \qquad \mathrm{Tr}(\rho_0) = 1 \tag{2}$$

(real or complex).
Coherence / Entropy (RC core) → iterative operator:

$$\rho_{t+1} = \Lambda_\theta(\rho_t,\, h,\, \mathrm{meta}) \tag{3}$$

Halt via learned policy or metric threshold.

2) State, metrics, and operators

Entropies Shannon (on probabilities p):

$$H(p) = -\sum_i p_i \log p_i \tag{4}$$

von Neumann (on state ρ):

$$S(\rho) = -\,\mathrm{Tr}(\rho \log \rho) \tag{5}$$

Coherence (two common choices) Relative entropy of coherence:

$$C_{\mathrm{rel}}(\rho) = S(\rho_{\mathrm{diag}}) - S(\rho) \tag{6}$$

ℓ_1-coherence:

$$C_{\ell_1}(\rho) = \sum_{i \neq j} |\rho_{ij}| \tag{7}$$

Task-aware free energy With POVM $\{M_j\}$,

$$E(\rho) \approx -\log \mathrm{Tr}(M_y\, \rho), \qquad F_T(\rho) = E(\rho) - T\, C_{\mathrm{rel}}(\rho)\,, \tag{8}$$

where T trades accuracy vs. coherence.

RC update (examples) *Gradient-like descent* with projection Π (to PSD, unit-trace):

$$\rho_{t+1} = \Pi[\rho_t - \eta\, \nabla_\rho (E(\rho_t) - T\, C_{\mathrm{rel}}(\rho_t))] \tag{9}$$

Learned map (CPTP-like):

$$\rho_{t+1} = \sum_k A_k(\phi)\, \rho_t\, A_k(\phi)^\top, \qquad \sum_k A_k^\top A_k = I \tag{10}$$

where ϕ summarizes h, step t, and meta.

Halting Metric stop: $|S(\rho_{t+1}) - S(\rho_t)| < \varepsilon$ or $\Delta F_T < \varepsilon$. Learned stop: $\pi_{\mathrm{halt}}(\rho_t, h)$.

Readout to logits

$$p(j) = \mathrm{Tr}(M_j \rho_T), \qquad \sum_j M_j = I, \quad M_j \succeq 0 \tag{11}$$

(In practice, a linear map $\mathrm{vec}(\rho_T) \to$ logits also works.)

Paper-ready fusion readout (single-line, boxed)

$$\boxed{u_t = \mathrm{vec}(\rho_t^\star)^\top W_r + V h_t,\ p_\theta(x_{t+1}) = \mathrm{softmax}(W_o\, u_t)} \tag{12}$$

where $\rho_t^\star$ is the converged/last RC state at step t.

Notation

Symbol	Type / Shape	Meaning
$x_{1:n}$	tokens	Input token sequence.
h	$\mathbb{R}^d$ (or $\mathbb{C}^d$)	Transformer hidden representation (pooled or per-step summary used by RC).
$P_O,\ P_S$	$\mathbb{R}^{d\times d}$	Learned projections for **objective** and **subjective** views.
$\psi^{(O)},\ \psi^{(S)}$	$\mathbb{R}^d$	Projected views: $\psi^{(O)} = P_O h,\ \psi^{(S)} = P_S h$.
α	$[0,1]$	Mixture weight for density assembly.
ρ	$\mathbb{R}^{d\times d}$ (PSD, $\mathrm{Tr}(\rho)=1$)	Density-like state; RC operates on ρ_t.
ρ_0	$\mathbb{R}^{d\times d}$	Init: $\rho_0 = \alpha\,\psi^{(O)}\psi^{(O)\top} + (1-\alpha)\,\psi^{(S)}\psi^{(S)\top}$.
U	$\mathbb{R}^{d\times r}$	Low-rank factor ($r \ll d$) with $\rho \approx UU^\top$.
M_j	$\mathbb{R}^{d\times d}$ (PSD)	Readout operators (POVM-like), $\sum_j M_j = I$.
I	$\mathbb{R}^{d\times d}$	Identity.
$p(j)$	$[0,1]$	Token probability: $p(j) = \mathrm{Tr}(M_j\,\rho_T)$.
u_t	$\mathbb{R}^m$	Fusion feature: $u_t = \mathrm{vec}(\rho_t^\star)^\top W_r + V h_t$.
W_r, V, W_o	matrices	Readout/fusion parameters; $p_\theta(x_{t+1}) = \mathrm{softmax}(W_o u_t)$.
Λ_θ	map	RC update: $\rho_{t+1} = \Lambda_\theta(\rho_t, h, \mathrm{meta})$.
$A_k(\phi)$	$\mathbb{R}^{d\times d}$	Kraus-like factors; $\sum_k A_k^\top A_k = I$.
ϕ	feature vector	Summary features for RC (e.g., h, t, meta).
$\Pi(\cdot)$	projection	Projection to the PSD, unit-trace set.
$E(\rho)$	scalar	Task-energy surrogate, e.g., $-\log \mathrm{Tr}(M_y\rho)$.
$S(\rho)$	scalar	von Neumann entropy: $S(\rho) = -\mathrm{Tr}(\rho \log \rho)$.
$C_{\mathrm{rel}}(\rho)$	scalar	Relative entropy of coherence: $S(\rho_{\mathrm{diag}}) - S(\rho)$.
$C_{\ell_1}(\rho)$	scalar	ℓ_1 coherence: $\sum_{i\neq j} \lvert\rho_{ij}\rvert$.
$F_T(\rho)$	scalar	Coherence-aware free energy: $E(\rho) - T\,C_{\mathrm{rel}}(\rho)$.
T	$\mathbb{R}_{\geq 0}$	Temperature.
η	$\mathbb{R}_{>0}$	Step size.
ε	$\mathbb{R}_{>0}$	Halt tolerance.
π_{halt}	policy	Learned halting head $\pi_{\mathrm{halt}}(\rho_t, h)$.
$t,\ T$	ints	RC step index and max steps.
$d,\ r$	ints	Hidden dim and low-rank factor ($r \ll d$).
$\mathrm{vec}(\cdot)$	operator	Vectorization of a matrix.

Basis note. ρ_{diag} is taken in a fixed computational basis aligned with the readout.

3) Training objectives

Base NLL

$$L_{\text{NLL}} = -\sum_t \log p_\theta(x_t \mid x_{<t}) \tag{13}$$

RC regularizers Free-energy schedule:

$$L_{\text{FE}} = \sum_t F_T(\rho_t) \tag{14}$$

Monotonicity (encourage $S \downarrow$, $C_{\text{rel}} \uparrow$):

$$L_{\text{mono}} = \sum_t \Big[\max(0, S(\rho_{t+1}) - S(\rho_t)) + \max(0, C_{\text{rel}}(\rho_t) - C_{\text{rel}}(\rho_{t+1})) \Big] \tag{15}$$

Optional O/S supervision (DPO/IPO-style); stability/CPTP penalties.

Total loss (example)

$$L = L_{\text{NLL}} + \lambda_{\text{FE}} L_{\text{FE}} + \lambda_{\text{mono}} L_{\text{mono}} + \lambda_{\text{stab}} L_{\text{stab}} + \lambda_{O/S} L_{O/S}\,. \tag{16}$$

4) Forward pass (inference) — minimal loop

1. Encode $h = \text{Transformer}(x_{1:n})$.
2. Dual heads: $(\psi^{(O)}, \psi^{(S)}) = (P_O h, P_S h)$; normalize.
3. Initialize: $\rho_0 = \alpha\, \psi^{(O)} {\psi^{(O)}}^\top + (1-\alpha)\, \psi^{(S)} {\psi^{(S)}}^\top$.
4. RC loop ($t = 0, \ldots, T-1$): $\rho_{t+1} = \Lambda_\theta(\rho_t, h, \text{meta})$; check halt.
5. Readout: *fusion* eq. above; decode via softmax.

5) Where RC attaches (three useful placements)

- **Outer loop (default):** after final layer on a compact summary of h (lowest latency hit).
- **Interleaved blocks:** a small RC step every k Transformer layers (good for long-context reasoning).
- **Head-only:** RC on a pooled token (e.g., BOS) for global planning (cheap/effective).

6) Data signals for O/S heads

Objective seeds: retrieval/citations, executable code tests, unit proof checks.
Subjective seeds: style, stance, hedging, valence, ambiguity in instructions.
Weak supervision: self-play pairs (more/less definitive), RAG correctness checks, programmatic detectors.

7) Evaluation (beyond standard benchmarks)

- ΔS per step: distribution of $S(\rho_t) - S(\rho_{t+1})$.
- Coherence lift: $C_{\mathrm{rel}}(\rho_T) - C_{\mathrm{rel}}(\rho_0)$.
- Quality vs. steps: accuracy/BLEU/Pass@k vs. T (elbow/auto-halt detection).
- Stability: PSD violations, trace drift; variance under small prompt perturbations.
- Thermo lens: do F_T trajectories correlate with correctness?

8) Scaling & systems

Cost control: low-rank $\rho = UU^\top$, $U \in \mathbb{R}^{d\times r}$, $r \ll d$; update $U \Rightarrow O(dr)$ memory.
Batching: share KV-cache from engine; RC loop is a lightweight MLP/Kraus mixer.
Latency knobs: $T_{\max}$, temperature T, rank r; enable early-halt.
Quantization: keep engine int8/int4; run RC in bf16/fp16 for numerical stability.

9) Minimal math to "summarize the whole thing"

(A) Density assembly

$$\rho_0 = \alpha\, \psi^{(O)} {\psi^{(O)}}^\top + (1-\alpha)\, \psi^{(S)} {\psi^{(S)}}^\top, \qquad \mathrm{Tr}(\rho_0) = 1 \tag{17}$$

(B) Coherence-aware free-energy descent (one step)

$$\rho_{t+1} = \Pi\Big[\rho_t - \eta\, \nabla_\rho \big(\underbrace{E(\rho_t)}_{\text{task loss}} - \underbrace{T\, C_{\mathrm{rel}}(\rho_t)}_{\text{coherence}} \big)\Big] \tag{18}$$

(C) Readout to token probabilities

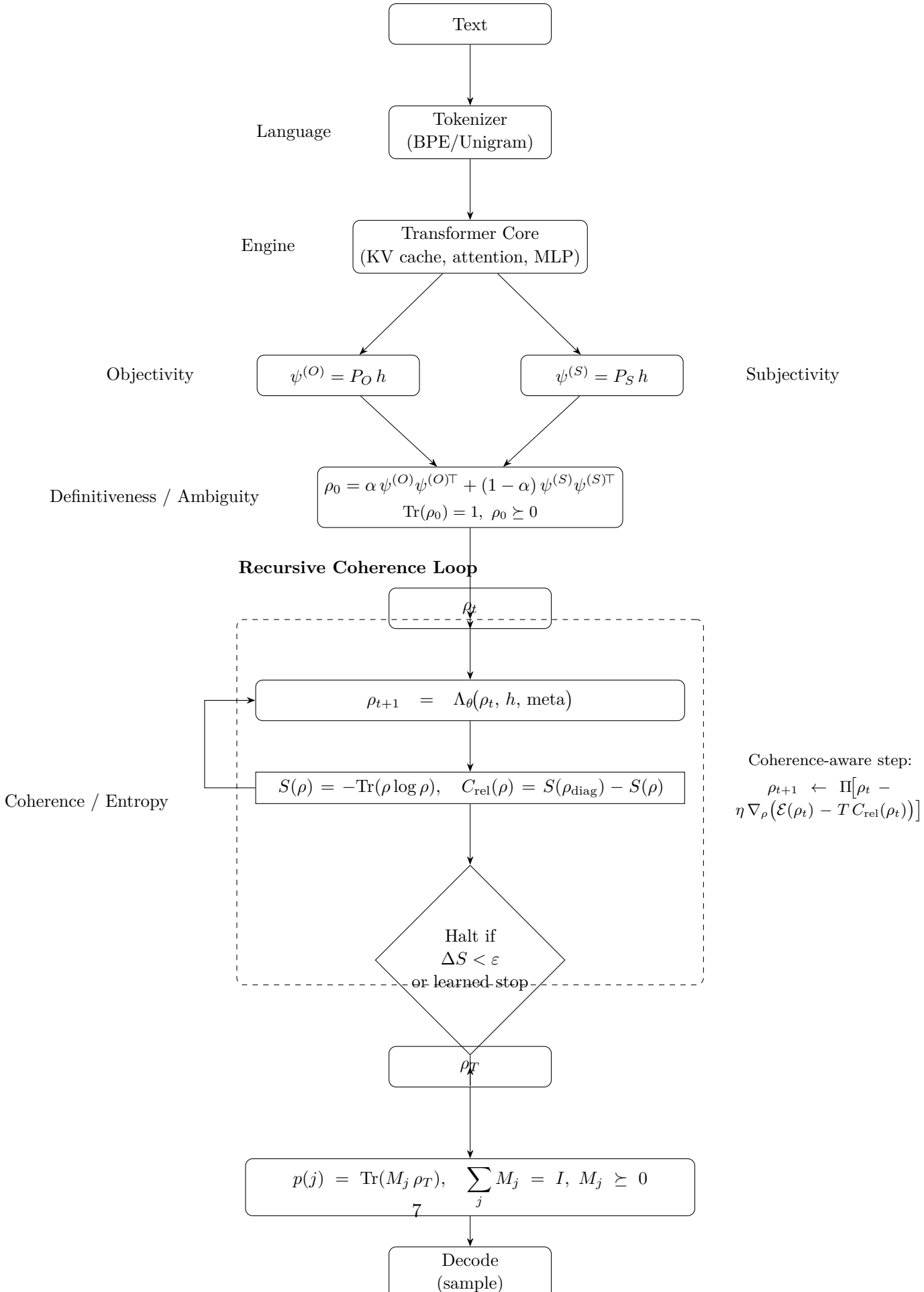
$$p(j) = \mathrm{Tr}(M_j\, \rho_T), \qquad \sum_j M_j = I, \quad M_j \succeq 0 \tag{19}$$

10) Practical implementation sketch (PyTorch)

1. Compute h with your Transformer.
2. Two linear heads $\to \psi^{(O)}$, $\psi^{(S)}$; normalize.
3. Rank-r construction: $U = [\, \alpha\, \psi^{(O)},\, (1-\alpha)\, \psi^{(S)} \,]$ (+ optional learned projector to widen rank).

4. RC block: tiny MLP $\rightarrow K$ Kraus-like factors (or ΔU); project to PSD and trace-1.
5. Learned halt head; cap $T_{\max}$.
6. Fusion readout (boxed): tie/untie W_o with embeddings as desired.

Engineering tip. Keep RC outside the base network to switch RC on/off, tune T, cap steps $T_{\max}$, and manage latency/quality trade-offs.


Text
Language
Tokenizer
(BPE/Unigram)
Engine
Transformer Core
(KV cache, attention, MLP)
Objectivity
$\psi^{(O)} = P_O\, h$
$\psi^{(S)} = P_S\, h$
Subjectivity
Definitiveness / Ambiguity
$\rho_0 = \alpha\, \psi^{(O)}\psi^{(O)\top} + (1-\alpha)\, \psi^{(S)}\psi^{(S)\top}$
$\mathrm{Tr}(\rho_0) = 1,\ \rho_0 \succeq 0$
Recursive Coherence Loop
ρ_t
$\rho_{t+1} \quad = \quad \Lambda_\theta\big(\rho_t,\, h,\, \mathrm{meta}\big)$
Coherence / Entropy
$S(\rho) \,=\, -\mathrm{Tr}(\rho \log \rho), \quad C_{\mathrm{rel}}(\rho) \,=\, S(\rho_{\mathrm{diag}}) - S(\rho)$
Coherence-aware step:
$\rho_{t+1} \ \leftarrow\ \Pi\big[\rho_t\ -$
$\eta\, \nabla_\rho \big(\mathcal{E}(\rho_t) - T\, C_{\mathrm{rel}}(\rho_t)\big)\big]$
Halt if
$\Delta S < \varepsilon$
or learned stop
ρ_T
$p(j) \ = \ \mathrm{Tr}(M_j\, \rho_T), \quad \sum_j M_j \ = \ I,\ M_j \ \succeq \ 0$
Decode
(sample)

References

- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., Polosukhin, I., "Attention Is All You Need." (NeurIPS 2017)
- Su, J., Lu, Y., Pan, S., "RoFormer: Enhanced Transformer with Rotary Position Embedding." (NeurIPS 2021)
- Press, O., Smith, N., Lewis, M., "Train Short, Test Long: Attention with Linear Biases (ALiBi)." (arXiv 2021)
- Dao, T., Fu, D. Y., Ermon, S., Rudra, A., Ré, C., "FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness." (NeurIPS 2022)
- Shazeer, N., "Fast Transformer Decoding: One Write-Head Is All You Need (Multi-Query Attention)." (arXiv 2019)
- Ainslie, J., Ontañón, S., Alberti, C., Cvicek, V., Fisher, Z., Pham, P., *et al.*, "GQA: Training Generalized Multi-Query Transformer Models Efficiently." (arXiv 2023)
- Fedus, W., Zoph, B., Shazeer, N., "Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity." (arXiv 2021; JMLR 2022)
- Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., *et al.*, "Training Compute-Optimal Large Language Models." (NeurIPS 2022; "Chinchilla")
- Zhang, B., Sennrich, R., "Root Mean Square Layer Normalization." (NeurIPS 2019 Workshop / arXiv 2019)
- Wang, S., Li, B., Khabsa, M., Fang, H., Ma, H., "DeepNet: Scaling Transformers to 1,000 Layers." (ICML 2022) (DeepNorm)
- Yang, G., Zhang, T., Saxe, A., "Tensor Programs V: Tuning Large Neural Networks via Zero-Shot Hyperparameter Transfer." (NeurIPS 2022) (μP)
- Kwon, W., Shao, R., Xie, Z., Xu, F. F., *et al.*, "Efficient Memory Management for Large Language Model Serving with PagedAttention (vLLM)." (SOSP 2023 / arXiv 2023)
- Chen, J., Sun, X., Zhang, J., Li, H., "Extending Context Window of Large Language Models via Positional Interpolation." (arXiv 2023)
- Lewis, P., Perez, E., Piktus, A., Petroni, F., *et al.*, "Retrieval-Augmented Generation for Knowledge-Intensive NLP." (NeurIPS 2020)
- Chen, T., Levkovitch, N., Benaim, S., *et al.*, "Accelerating Large Language Model Decoding with Speculative Sampling/Decoding." (arXiv 2023) (Any equivalent "speculative decoding" paper is fine; multiple variants exist.)
- Ouyang, L., Wu, J., Jiang, X., Almeida, D., *et al.*, "Training Language Models to Follow Instructions with Human Feedback." (NeurIPS 2022) (RLHF)

- Rafailov, R., Sharma, A., Mitchell, E., Ermon, S., Finn, C., "Direct Preference Optimization: Your Language Model Is Secretly a Reward Model." (NeurIPS 2023) (DPO)
- Hong, J., Kim, S., Na, B., *et al.*, "ORPO: Monolithic Preference Optimization without a Separate Reward Model." (EMNLP 2024)
- Nielsen, M. A., Chuang, I. L., *Quantum Computation and Quantum Information*. (Cambridge University Press, 2010) (density matrices, POVMs, CPTP/Kraus)
- Baumgratz, T., Cramer, M., Plenio, M. B., "Quantifying Coherence." (Physical Review Letters, 2014) (relative-entropy and ℓ_1 coherence)
- Higham, N. J., "Computing the Nearest Correlation Matrix—A Problem from Finance." (IMA Journal of Numerical Analysis, 2002) (PSD projection ideas)
- Wang, W., Carreira-Perpiñán, M. Á., "Projection onto the Probability Simplex: An Efficient Algorithm with a Simple Proof." (arXiv 2013) (simplex/trace-1 projection)
- (Definition reference) "Von Neumann Entropy." (Standard definition; e.g., Wikipedia or any quantum information textbook)
- Liu, Jian-wei; Xu, Bing-rong; Song, Zhi-yan, A Survey of Recursive and Recurrent Neural Networks. arXiv:2510.17867 [cs.NE], 16 Oct 2025