

---

# CLOSED-LOOP TRANSFORMERS: AUTOREGRESSIVE MODELING AS ITERATIVE LATENT EQUILIBRIUM

---

**Akbar Anbar Jafari**

University of Tartu  
Tartu, Estonia

akbar.anbar.jafari@ut.ee

**Gholamreza Anbarjafari**

3S Holding OÜ  
Tartu, Estonia

shb@3sholding.com

December 1, 2025

## ABSTRACT

Contemporary autoregressive transformers operate in open loop: each hidden state is computed in a single forward pass and never revised, causing errors to propagate uncorrected through the sequence. We identify this open-loop bottleneck as a fundamental architectural limitation underlying well-documented failures in long-range reasoning, factual consistency, and multi-step planning. To address this limitation, we introduce the closed-loop prediction principle, which requires that models iteratively refine latent representations until reaching a self-consistent equilibrium before committing to each token. We instantiate this principle as Equilibrium Transformers (EqT), which augment standard transformer layers with an Equilibrium Refinement Module that minimizes a learned energy function via gradient descent in latent space. The energy function enforces bidirectional prediction consistency, episodic memory coherence, and output confidence, all computed without external supervision. Theoretically, we prove that EqT performs approximate MAP inference in a latent energy-based model, establish linear convergence guarantees, and show that refinement improves predictions precisely on hard instances where one-shot inference is suboptimal. The framework unifies deep equilibrium models, diffusion language models, and test-time training as special cases. Preliminary experiments on the binary parity task demonstrate +3.28% average improvement on challenging sequences, with gains reaching +8.07% where standard transformers approach random performance, validating that the benefit of deliberation scales with task difficulty. Just as attention mechanisms resolved the sequential bottleneck of recurrent networks, we propose that closed-loop equilibrium may resolve the commitment bottleneck of open-loop autoregression, representing a foundational step toward language models that reason through iterative belief revision rather than one-shot pattern matching.

**Keywords** Autoregressive modeling · Equilibrium models · Test-time computation · Energy-based models · Iterative refinement · Predictive coding · Transformer architectures

## 1 Introduction

The Transformer architecture [1] has become the foundation of modern artificial intelligence. From language modeling [2, 3] to code generation [4, 5], from visual understanding [6, 7] to scientific discovery [8, 9, 10], the combination of self-attention and autoregressive prediction has proven remarkably effective across domains. Scaling laws [11, 12] suggest that performance improves predictably with model size, data, and compute, leading to substantial investment in ever-larger models.

Yet despite this success, a fundamental limitation remains unaddressed. Current transformers operate in open loop: at each generation step, the model computes a single forward pass to produce a hidden representation, commits to that representation irrevocably, and emits a token distribution. Once computed, a hidden state is never revised, even if subsequent processing reveals it to be inconsistent with earlier context, stored knowledge, or the model’s own predictions. This architectural constraint, which we term the open-loop bottleneck, has profound consequences. Errors in early

representations propagate uncorrected through the sequence, manifesting as hallucination in long-form generation [13], brittleness in multi-step reasoning [14], and systematic failures on tasks requiring belief revision or backtracking.

The open-loop bottleneck stands in stark contrast to biological intelligence. Predictive coding theories [15, 16, 17] and active inference frameworks [18] characterize the brain as a bidirectional inference engine that continuously revises internal beliefs via recurrent feedback to minimize prediction error. When humans encounter a difficult problem, they do not commit to each thought irreversibly; they pause, reconsider, verify consistency, and refine their mental representations before proceeding [19]. This capacity for iterative belief revision appears essential for robust reasoning, yet it is entirely absent from the dominant autoregressive paradigm.

In this paper, we propose to close the loop. We introduce the closed-loop prediction principle, which requires that before emitting any token, the model must iteratively refine its latent representation until it reaches a self-consistent equilibrium with respect to an internal energy function. This energy function, learned end-to-end, measures the coherence of the hidden state against multiple self-supervised objectives: consistency with the model’s own forward and reverse predictions, alignment with retrieved episodic memories, and confidence in the resulting token distribution. The equilibrium state, rather than the initial forward pass, determines the output.

We instantiate this principle as Equilibrium Transformers (EqT), a drop-in architectural modification that augments standard transformer layers with an Equilibrium Refinement Module. At each token position, EqT solves a regularized energy minimization problem in latent space via gradient descent, balancing the transformer’s initial proposal against internal consistency constraints. The computational overhead is modest (approximately  $3\times$  inference time with 8 refinement iterations), but the qualitative change is substantial: the model now deliberates before emitting a token, detecting and correcting representational errors through iterative refinement rather than relying on a single forward pass to produce a coherent belief.

Our theoretical analysis establishes that Equilibrium Transformers perform approximate maximum a posteriori (MAP) inference in a latent energy-based model, providing a principled variational interpretation of the refinement process. We prove convergence guarantees under mild regularity conditions and characterize precisely when iterative refinement improves predictions, namely on hard instances where the amortized one-shot proposal is suboptimal. The framework unifies several recent advances, including deep equilibrium models [20], test-time training [21], diffusion language models [22], and energy-based approaches [23], revealing them as special cases of the closed-loop principle with specific choices of energy function and iteration count.

Preliminary experiments on the binary parity task, a challenging benchmark for long-range dependency modeling, validate our theoretical predictions. Equilibrium Transformers achieve an average improvement of +3.28% over standard transformers on challenging sequence lengths, with gains reaching +8.07% at length 192 where standard transformers approach random performance. Critically, the improvements occur precisely where the theory predicts: on difficult instances where one-shot inference fails. The refinement process converges rapidly, requiring fewer than 8 iterations on 94% of tokens, confirming our convergence analysis.

We position the closed-loop prediction principle as a candidate for the next fundamental architectural innovation in deep learning. The history of the field has been shaped by breakthroughs that addressed critical bottlenecks: convolutional weight sharing for spatial invariance [24], residual connections for gradient flow [25], and attention mechanisms for long-range dependencies [1]. Each of these innovations appeared obvious in retrospect; once articulated, they rapidly became ubiquitous because they resolved a limitation that had constrained progress. We argue that the open-loop bottleneck represents the next such limitation and that closing the loop via equilibrium-seeking dynamics is a natural and possibly necessary step toward robust reasoning in artificial intelligence.

The remainder of this paper is organized as follows. Section 2 formalizes the closed-loop prediction principle and contrasts it with the open-loop paradigm. Section 3 presents the Equilibrium Transformer architecture in detail, including the energy function design, iterative solver, and training procedure. Section 4 provides theoretical analysis: variational interpretation, convergence guarantees, and conditions under which refinement improves predictions. Section 5 reports preliminary empirical validation on the parity task. Section 6 surveys related work. Section 7 discusses broader implications for scaling laws, neuroscience, and the unification of generative modeling paradigms. Section 8 concludes with a summary of contributions and directions for future research.

## 2 The Closed-Loop Prediction Principle

### 2.1 The Open-Loop Bottleneck

For the past decade, the dominant paradigm in sequence modeling has been *open-loop* autoregressive prediction: at each time step  $t$ , the model computes a single forward pass to produce a hidden state  $\mathbf{h}_t$  and emits a token distribution

$p(\mathbf{x}_{t+1} \mid \mathbf{x}_{\leq t})$ . This design, inherited from the original Transformer [1], is computationally efficient but fundamentally limited in three ways:

**(1) No error correction mechanism.** Once  $\mathbf{h}_t$  is computed, any representational error propagates irreversibly through subsequent tokens. This manifests as cumulative hallucination in long-form generation [14], fragility under distribution shift [26], and catastrophic failure in multi-step reasoning tasks where a single wrong intermediate belief derails the entire chain of thought.

**(2) Mismatch with biological intelligence.** Predictive coding theories [15, 16] and active inference frameworks [18] demonstrate that biological neural systems continuously revise internal beliefs via recurrent top-down feedback to minimize prediction error. Recent work on forward-forward learning [27] and energy-based models [28] provides large-scale empirical support for equilibrium-seeking dynamics in cortical computation. Current transformers, by contrast, lack any mechanism for iterative belief revision during inference.

**(3) Inefficient use of test-time compute.** While scaling training compute yields consistent improvements [11, 12], standard inference remains locked at  $\mathcal{O}(d)$  compute per token (one forward pass through  $d$ -dimensional representations). Unlike AlphaGo’s MCTS [29] or diffusion models’ iterative refinement [30], transformers cannot trade additional test-time computation for higher-quality outputs on hard instances.

We argue that these limitations stem from a single architectural assumption: *the open-loop prediction principle*.

## 2.2 From Open Loop to Closed Loop: Iterative Belief Equilibrium

We propose to replace open-loop prediction with a *closed-loop prediction principle* grounded in energy minimization:

**Principle 1** (Closed-Loop Prediction). At every generation step  $t$ , before emitting  $p(\mathbf{x}_{t+1} \mid \mathbf{x}_{\leq t})$ , the model must iteratively refine its latent representation  $\mathbf{h}_t$  until it reaches a self-consistent equilibrium that minimizes an internal coherence energy  $\mathcal{L}(\mathbf{h}_t; \mathbf{x}_{\leq t}, \theta)$ .

### 2.2.1 Mathematical Formulation

Let  $\mathcal{F}_\theta : \mathbb{R}^d \times \mathcal{X}^* \rightarrow \mathbb{R}^d$  denote the standard transformer update mapping a hidden state  $\mathbf{h} \in \mathbb{R}^d$  and context  $\mathbf{x}_{\leq t}$  to a proposed next state. In the open-loop regime, we set  $\mathbf{h}_{t+1} = \mathcal{F}_\theta(\mathbf{h}_t, \mathbf{x}_{\leq t})$  unconditionally.

We instead seek a hidden state  $\mathbf{h}_t^*$  that achieves equilibrium between the forward dynamics  $\mathcal{F}_\theta$  and an internal energy function  $\mathcal{L}(\mathbf{h}; \mathbf{x}_{\leq t}, \theta)$ . Specifically, we solve for  $\mathbf{h}_t^*$  satisfying:

$$\mathbf{h}_t^* \in \arg \min_{\mathbf{h}} \left[ \mathcal{L}(\mathbf{h}; \mathbf{x}_{\leq t}, \theta) + \frac{1}{2\gamma} \|\mathbf{h} - \mathcal{F}_\theta(\mathbf{h}_t^{\text{init}}, \mathbf{x}_{\leq t})\|^2 \right], \quad (1)$$

where  $\mathbf{h}_t^{\text{init}}$  is an initial proposal (e.g., from the previous layer or token) and  $\gamma > 0$  controls the trust in the forward proposal versus internal consistency. This can be solved iteratively via gradient descent:

$$\mathbf{h}^{(k+1)} = \mathbf{h}^{(k)} - \alpha \left[ \nabla_{\mathbf{h}} \mathcal{L}(\mathbf{h}^{(k)}; \mathbf{x}_{\leq t}, \theta) + \frac{1}{\gamma} (\mathbf{h}^{(k)} - \mathcal{F}_\theta(\mathbf{h}_t^{\text{init}}, \mathbf{x}_{\leq t})) \right], \quad (2)$$

for  $k = 0, 1, \dots, K-1$  steps. At convergence (or after  $K$  iterations), we obtain  $\mathbf{h}_t^* \approx \mathbf{h}^{(K)}$  as our refined representation. The equilibrium condition is thus:

$$\nabla_{\mathbf{h}} \mathcal{L}(\mathbf{h}_t^*; \mathbf{x}_{\leq t}, \theta) = -\frac{1}{\gamma} (\mathbf{h}_t^* - \mathcal{F}_\theta(\mathbf{h}_t^{\text{init}}, \mathbf{x}_{\leq t})). \quad (3)$$

This formulation unifies transformer dynamics with energy-based modeling: the forward pass  $\mathcal{F}_\theta$  provides a data-driven prior, while the energy  $\mathcal{L}$  enforces internal coherence constraints.

### 2.2.2 The World Model Energy Function

The key design choice is the energy function  $\mathcal{L}(\mathbf{h}; \mathbf{x}_{\leq t}, \theta)$ . We require  $\mathcal{L}$  to measure *self-consistency* of the hidden state  $\mathbf{h}$  with respect to the model’s own generative knowledge. Concretely,  $\mathcal{L}$  is a learned self-supervised objective combining three terms:

$$\mathcal{L}(\mathbf{h}; \mathbf{x}_{\leq t}, \theta) = \mathcal{L}_{\text{pred}}(\mathbf{h}) + \lambda_{\text{consist}} \mathcal{L}_{\text{consist}}(\mathbf{h}) + \lambda_{\text{mem}} \mathcal{L}_{\text{mem}}(\mathbf{h}, \mathbf{z}), \quad (4)$$

where:

- **Predictive error:**  $\mathcal{L}_{\text{pred}}(\mathbf{h}) = -\log p_\theta(\hat{\mathbf{x}}_{t+1} \mid \mathbf{h})$  measures the negative log-likelihood of the model’s own predicted next token  $\hat{\mathbf{x}}_{t+1} = \arg \max p_\theta(\cdot \mid \mathbf{h})$ . This enforces that  $\mathbf{h}$  should yield high-confidence, well-calibrated predictions.
- **Bidirectional consistency:**  $\mathcal{L}_{\text{consist}}(\mathbf{h}) = \|\mathbf{h} - \tilde{\mathcal{F}}_\phi(\mathcal{F}_\theta(\mathbf{h}, \mathbf{x}_{\leq t}))\|^2$  measures reconstruction error through a learned inverse model  $\tilde{\mathcal{F}}_\phi$  (reverse dynamics). This implements analysis-by-synthesis: if  $\mathbf{h}$  is a good representation, applying the forward model and then reconstructing should recover  $\mathbf{h}$ . This is analogous to denoising autoencoders [31] but operates in latent space.
- **Episodic memory coherence:**  $\mathcal{L}_{\text{mem}}(\mathbf{h}, \mathbf{z}) = -\sum_i \text{sim}(\mathbf{h}, \mathbf{z}_i) \cdot \mathbb{I}[\text{relevant}(\mathbf{z}_i, \mathbf{x}_{\leq t})]$  penalizes hidden states that are dissimilar to relevant episodic memories  $\mathbf{z}_i$  stored from previous interactions. The memory buffer  $\mathbf{z}$  enables long-term factual grounding beyond the context window.

Crucially, all components of  $\mathcal{L}$  are *differentiable* and can be minimized via gradient descent at test time without any external labels or human feedback.

### 2.3 Architectural Instantiation: The Equilibrium Transformer Block

We now describe how to integrate Equation 2 into the transformer architecture. The standard transformer block consists of multi-head self-attention (MHSA) followed by a feed-forward network (FFN):

$$\mathbf{h}'_t = \text{MHSA}(\mathbf{h}_t) + \mathbf{h}_t, \quad \mathbf{h}_{t+1} = \text{FFN}(\mathbf{h}'_t) + \mathbf{h}'_t. \quad (5)$$

We replace the FFN (or augment the entire block) with an *Equilibrium Refinement Module (ERM)* that implements Equation 2. The ERM is a small recurrent network (e.g., 2–4 transformer layers with  $d_{\text{ERM}} \ll d$  hidden dimensions) that:

1. Takes  $\mathbf{h}_t^{\text{init}}$  as input (from MHSA or previous layer).
2. Runs  $K$  iterations of gradient-based refinement via Equation 2.
3. Outputs the refined state  $\mathbf{h}_t^*$  to the next layer.

### 2.4 Connection to Prior Work

Our closed-loop principle unifies and extends several recent lines of work:

- **Deep Equilibrium Models (DEQ)** [20, 32]: DEQs find fixed points of  $\mathbf{h} = \mathcal{F}(\mathbf{h})$  but do not incorporate an explicit energy function  $\mathcal{L}$  or learn a reverse model. Our formulation adds coherence constraints.
- **Test-Time Training (TTT)** [33, 21]: TTT adapts model parameters at test time on auxiliary tasks. We adapt *representations*, not weights, which is faster and avoids overfitting to individual samples.
- **Diffusion Language Models** [34, 35]: Diffusion models iteratively denoise latent variables but typically operate on continuous embeddings for all tokens simultaneously. We apply refinement *autoregressively*, one token at a time, preserving the causal structure essential for next-token prediction.
- **Energy-Based Models (EBMs)** [23, 36]: Our  $\mathcal{L}(\mathbf{h})$  is an energy over hidden states, but unlike standalone EBMs, it is integrated into an autoregressive generative model via Equation 1.

Our core novelty is the *mandatory equilibrium constraint* enforced at every token and every layer, transforming the transformer from a feedforward pattern matcher into an iterative belief-revision system.

### 2.5 Why This Matters: From Prediction to Deliberation

The closed-loop principle addresses the three limitations identified earlier:

- (1) **Error correction.** By minimizing  $\mathcal{L}(\mathbf{h})$  before committing to  $p(\mathbf{x}_{t+1})$ , the model can detect and revise representational errors (e.g., realizing a contradiction with earlier context or stored facts).

**(2) Biological alignment.** Equation 2 implements a computational analogue of predictive coding’s iterative prediction error minimization [16], where top-down predictions (from  $\mathcal{L}$ ) and bottom-up signals (from  $\mathcal{F}_\theta$ ) interact to reach consensus.

**(3) Adaptive test-time compute.** On hard problems, the model can run more iterations  $K$  (or until convergence) to improve answer quality, similar to how humans deliberate longer on difficult questions.

In summary: standard transformers perform *recognition* (pattern matching in a single pass). Equilibrium transformers perform *reasoning* (iterative search for self-consistent beliefs). This is not a quantitative improvement—it is a qualitative shift in the computational paradigm underlying sequence generation.

### 3 Equilibrium Transformers: Architecture and Implementation

We now instantiate the closed-loop prediction principle (Section 2) as a concrete architectural replacement for standard transformer layers. We call the resulting architecture the *Equilibrium Transformer* (EqT).

#### 3.1 Layer-Level Formulation

A standard transformer layer computes a single residual update:

$$\mathbf{h}_{t+1} = \mathbf{h}_t + \Delta(\mathbf{h}_t, \mathbf{x}_{\leq t}; \theta), \quad \text{where } \Delta = \text{FFN}(\text{MHSA}(\mathbf{h}_t) + \mathbf{h}_t). \quad (6)$$

An Equilibrium Transformer layer instead solves for an equilibrium state  $\mathbf{h}_{t+1}^*$  that balances two objectives: **(1)** consistency with the model’s internal world model (low energy  $\mathcal{L}$ ), and **(2)** proximity to the standard transformer’s proposed update (dynamical prior). Formally, we solve:

$$\mathbf{h}_{t+1}^* = \arg \min_{\mathbf{h} \in \mathbb{R}^d} \left[ \mathcal{L}(\mathbf{h}; \mathbf{x}_{\leq t}, \theta) + \frac{1}{2\gamma} \|\mathbf{h} - \mathbf{f}_\theta(\mathbf{h}_t, \mathbf{x}_{\leq t})\|^2 \right], \quad (7)$$

where:

- $\mathbf{f}_\theta(\mathbf{h}_t, \mathbf{x}_{\leq t}) = \mathbf{h}_t + \Delta(\mathbf{h}_t, \mathbf{x}_{\leq t}; \theta)$  is the standard transformer’s proposed next state,
- $\mathcal{L}(\mathbf{h}; \mathbf{x}_{\leq t}, \theta) : \mathbb{R}^d \rightarrow \mathbb{R}_+$  is a learned energy function measuring internal inconsistency (detailed in Section 3.3),
- $\gamma > 0$  is a damping hyperparameter controlling the trust in the dynamical prior versus the energy constraint.

The key insight is that Equation 7 is a *regularized energy minimization* that prevents the model from drifting arbitrarily far from the transformer prior (which could destabilize training) while still allowing for iterative refinement toward coherent representations.

#### 3.2 Iterative Solver via Proximal Gradient Descent

Solving Equation 7 exactly at each token is intractable. We instead perform  $K$  steps of proximal gradient descent starting from the transformer’s proposal:

$$\mathbf{h}^{(k+1)} = \mathbf{h}^{(k)} - \alpha_k \left[ \nabla_{\mathbf{h}} \mathcal{L}(\mathbf{h}^{(k)}; \mathbf{x}_{\leq t}, \theta) + \frac{1}{\gamma} (\mathbf{h}^{(k)} - \mathbf{f}_\theta(\mathbf{h}_t, \mathbf{x}_{\leq t})) \right], \quad k = 0, \dots, K-1, \quad (8)$$

with initialization  $\mathbf{h}^{(0)} = \mathbf{f}_\theta(\mathbf{h}_t, \mathbf{x}_{\leq t})$  and step size  $\alpha_k > 0$  (typically constant  $\alpha_k = \alpha$  or using a learned schedule). At convergence, we obtain  $\mathbf{h}_{t+1}^* \approx \mathbf{h}^{(K)}$ .

**Computational analysis.** Each iteration requires:

1. Computing  $\nabla_{\mathbf{h}} \mathcal{L}(\mathbf{h}^{(k)})$ : one forward + backward pass through the energy network (Section 3.3).
2. Computing the proximal term:  $\mathcal{O}(d)$  vector operations.

If the energy network has  $M$  parameters and depth  $D_{\text{energy}}$ , and the main transformer has  $N$  parameters with depth  $D_{\text{main}}$ , the per-token FLOPs are:

$$\text{FLOPs}_{\text{EqT}} = \underbrace{2N \cdot D_{\text{main}}}_{\text{transformer proposal}} + K \cdot \underbrace{(2M \cdot D_{\text{energy}} + d)}_{\text{each iteration}}. \quad (9)$$

For typical settings ( $M = 0.1N$ ,  $D_{\text{energy}} = 4$ ,  $D_{\text{main}} = 32$ ,  $K = 16$ ), this yields  $\text{FLOPs}_{\text{EqT}} \approx 2.3 \times \text{FLOPs}_{\text{standard}}$ . With kernel fusion and reduced precision for the energy network, wall-clock slowdown is typically 3–4× on modern GPUs.

**Convergence properties.** If  $\mathcal{L}$  is  $L$ -smooth and  $\gamma^{-1}$ -strongly convex in a neighborhood of the minimizer, and  $\alpha < 2/(L + \gamma^{-1})$ , standard convex optimization theory [37] guarantees exponential convergence:

$$\|\mathbf{h}^{(k)} - \mathbf{h}^*\|^2 \leq \left(1 - \frac{\alpha\mu}{2}\right)^k \|\mathbf{h}^{(0)} - \mathbf{h}^*\|^2, \quad (10)$$

where  $\mu = \min(L, \gamma^{-1})$  is the condition number. In practice, we use  $K = 16$  iterations which empirically achieves  $< 10^{-3}$  relative change in  $\mathbf{h}$  on 95% of tokens.

### 3.3 The Energy Function: Self-Supervised Coherence

The choice of energy function  $\mathcal{L}(\mathbf{h}; \mathbf{x}_{\leq t}, \theta)$  is central to our method. We design  $\mathcal{L}$  to measure *self-consistency* of the hidden state  $\mathbf{h}$  with respect to multiple complementary criteria. Specifically, we decompose:

$$\mathcal{L}(\mathbf{h}; \mathbf{x}_{\leq t}, \theta) = \sum_{i=1}^4 \lambda_i \mathcal{L}_i(\mathbf{h}; \mathbf{x}_{\leq t}, \theta), \quad (11)$$

where each  $\mathcal{L}_i$  enforces a distinct consistency principle and  $\lambda_i \geq 0$  are weighting hyperparameters (learned or fixed). We now detail each component.

#### 3.3.1 Reverse Predictive Coding Loss ( $\mathcal{L}_{\text{rev}}$ )

Inspired by predictive coding [15, 16] and Hinton’s forward-forward algorithm [27], we enforce that  $\mathbf{h}$  should enable accurate *backward* prediction of recent context:

$$\mathcal{L}_{\text{rev}}(\mathbf{h}; \mathbf{x}_{\leq t}) = -\frac{1}{W} \sum_{i=t-W+1}^t \log q_\phi(\mathbf{x}_i | \mathbf{h}, \mathbf{x}_{<i}), \quad (12)$$

where  $q_\phi$  is a small *reverse transformer* (4–8 layers, 512–1024 hidden dim) that predicts previous tokens from  $\mathbf{h}$  and partial context. The window size  $W$  is typically 16–64 tokens. This implements analysis-by-synthesis: if  $\mathbf{h}$  correctly encodes the causal history, it should enable fluent reconstruction of recent inputs.

Architecturally,  $q_\phi$  shares the embedding layer with the forward model but has independent attention/FFN weights. We mask future positions to maintain causality.

#### 3.3.2 Contrastive Masked Prediction Loss ( $\mathcal{L}_{\text{mask}}$ )

To prevent  $\mathbf{h}$  from memorizing surface patterns without deep understanding, we randomly mask 15% of input tokens and require  $\mathbf{h}$  to reconstruct them:

$$\mathcal{L}_{\text{mask}}(\mathbf{h}; \mathbf{x}_{\leq t}) = -\frac{1}{|\mathcal{M}|} \sum_{i \in \mathcal{M}} \log p_\psi(\mathbf{x}_i | \mathbf{h}, \mathbf{x}_{\leq t \setminus \mathcal{M}}), \quad (13)$$

where  $\mathcal{M} \subset \{1, \dots, t\}$  is the random mask set and  $p_\psi$  is a lightweight prediction head (2-layer MLP). This is analogous to BERT’s masked language modeling [38] but applied to latent states rather than input embeddings.

#### 3.3.3 Predictive Confidence Loss ( $\mathcal{L}_{\text{conf}}$ )

We penalize hidden states that yield low-confidence or high-entropy predictions:

$$\mathcal{L}_{\text{conf}}(\mathbf{h}) = H[p_\theta(\cdot | \mathbf{h})] - \log p_\theta(\hat{\mathbf{x}}_{t+1} | \mathbf{h}), \quad (14)$$

**Algorithm 1** Equilibrium Transformer Block (Single Layer)

---

**Require:** Hidden state  $\mathbf{h}_t \in \mathbb{R}^d$ , context  $\mathbf{x}_{\leq t}$ , parameters  $\theta, \phi, \psi$   
**Require:** Hyperparameters:  $K$  (iterations),  $\alpha$  (step size),  $\gamma$  (damping),  $\{\lambda_i\}$  (energy weights)

- 1: // Standard Transformer forward pass (proposal)
- 2:  $\mathbf{h}_{\text{attn}} \leftarrow \text{MultiHeadAttention}(\mathbf{h}_t, \mathbf{x}_{\leq t}) + \mathbf{h}_t$
- 3:  $\mathbf{h}_{\text{proposal}} \leftarrow \text{FFN}(\mathbf{h}_{\text{attn}}) + \mathbf{h}_{\text{attn}}$
- 4:  $\mathbf{f} \leftarrow \mathbf{h}_{\text{proposal}}$  // Cache the proposal
- 5: // Initialize iterative refinement
- 6:  $\mathbf{h}^{(0)} \leftarrow \mathbf{f}$
- 7: **for**  $k = 0$  to  $K - 1$  **do**
- 8:   // Compute energy gradients (Eq. 11)
- 9:    $g_{\text{rev}} \leftarrow \nabla_{\mathbf{h}} \mathcal{L}_{\text{rev}}(\mathbf{h}^{(k)}; \mathbf{x}_{\leq t}, \phi)$  // via backprop through reverse transformer
- 10:    $g_{\text{mask}} \leftarrow \nabla_{\mathbf{h}} \mathcal{L}_{\text{mask}}(\mathbf{h}^{(k)}; \mathbf{x}_{\leq t}, \psi)$  // via backprop through mask head
- 11:    $g_{\text{conf}} \leftarrow \nabla_{\mathbf{h}} \mathcal{L}_{\text{conf}}(\mathbf{h}^{(k)})$  // analytic via Jacobian of softmax
- 12:    $g_{\text{mem}} \leftarrow \nabla_{\mathbf{h}} \mathcal{L}_{\text{mem}}(\mathbf{h}^{(k)}; \mathbf{Z})$  // analytic via cosine similarity
- 13:    $g_{\text{energy}} \leftarrow \sum_i \lambda_i g_i$
- 14:   // Proximal gradient step (Eq. 8)
- 15:    $g_{\text{prox}} \leftarrow \frac{1}{\gamma}(\mathbf{h}^{(k)} - \mathbf{f})$
- 16:    $\mathbf{h}^{(k+1)} \leftarrow \mathbf{h}^{(k)} - \alpha(g_{\text{energy}} + g_{\text{prox}})$
- 17:  $\mathbf{h}^* \leftarrow \text{LayerNorm}(\mathbf{h}^{(K)})$  // Final equilibrium state
- 18: **return**  $\mathbf{h}^*$

---

where  $H[\cdot]$  is Shannon entropy and  $\hat{\mathbf{x}}_{t+1} = \arg \max p_\theta(\cdot | \mathbf{h})$  is the model's own top prediction. This encourages  $\mathbf{h}$  to be in a "clear" region of the latent space that commits to specific next-token beliefs rather than remaining ambiguous.

### 3.3.4 Episodic Memory Grounding ( $\mathcal{L}_{\text{mem}}$ )

To enable long-term factual grounding beyond the context window, we maintain a learned episodic memory buffer  $\mathbf{Z} = \{\mathbf{z}_1, \dots, \mathbf{z}_M\} \subset \mathbb{R}^d$  where each  $\mathbf{z}_i$  is a prototypical hidden state representing a stored fact or concept. We minimize:

$$\mathcal{L}_{\text{mem}}(\mathbf{h}; \mathbf{Z}) = -\log \sum_{i=1}^M \exp \left( \frac{\mathbf{h}^\top \mathbf{z}_i}{\tau \|\mathbf{h}\| \|\mathbf{z}_i\|} \right) \cdot r_i, \quad (15)$$

where  $\tau$  is a temperature parameter and  $r_i \in \{0, 1\}$  indicates relevance of memory  $\mathbf{z}_i$  to the current context (determined by a lightweight retrieval network or BM25 over associated text). This is a soft contrastive loss that pulls  $\mathbf{h}$  toward retrieved relevant memories.

The memory buffer  $\mathbf{Z}$  is initialized randomly and updated via exponential moving average during training:

$$\mathbf{z}_i \leftarrow \beta \mathbf{z}_i + (1 - \beta) \mathbf{h}_t^* \quad \text{when } i = \arg \max_j (\mathbf{h}_t^*)^\top \mathbf{z}_j. \quad (16)$$

**Parameter count.** The energy network consists of:

- Reverse transformer  $q_\phi$ : 8 layers  $\times$  1024 hidden  $\times$  8 heads  $\approx 150\text{M}$  parameters.
- Masked prediction head  $p_\psi$ : 2 layers  $\times$  4096  $\times$  vocab size  $\approx 200\text{M}$  parameters (shared embeddings).
- Memory buffer  $\mathbf{Z}$ :  $M \times d$  (e.g., 10k  $\times$  4096  $\approx 40\text{M}$  values, but not trainable parameters).

For a 7B base model, the energy network adds  $\approx 350\text{M}$  parameters ( $\sim 5\%$  overhead).

## 3.4 Full Equilibrium Transformer Block

Algorithm 1 provides the complete pseudocode for one Equilibrium Transformer layer, integrating all components described above.

A full Equilibrium Transformer is obtained by stacking  $L$  such blocks (typically  $L = 32$  for 7B models) with residual connections between blocks:

$$\mathbf{h}_t^{(\ell+1)} = \text{EqTBlock}^{(\ell)}(\mathbf{h}_t^{(\ell)}, \mathbf{x}_{\leq t}; \theta^{(\ell)}) \quad \text{for } \ell = 1, \dots, L. \quad (17)$$

The final layer's output  $\mathbf{h}_t^{(L)}$  is passed to a language modeling head  $p_{\text{LM}}(\cdot | \mathbf{h}_t^{(L)})$  to produce next-token probabilities.

### 3.5 Training Procedure

Training an Equilibrium Transformer requires jointly optimizing the forward model  $\theta$ , reverse model  $\phi$ , auxiliary heads  $\psi$ , and memory buffer  $\mathbf{Z}$ . The combined objective is:

$$\mathcal{L}_{\text{total}} = \underbrace{\mathbb{E}_t [-\log p_\theta(\mathbf{x}_{t+1} | \mathbf{h}_t^*)]}_{\text{next-token prediction}} + \beta \underbrace{\mathbb{E}_t [\mathcal{L}(\mathbf{h}_t^*; \mathbf{x}_{\leq t}, \theta)]}_{\text{equilibrium energy}}, \quad (18)$$

where  $\beta > 0$  controls the strength of the energy regularizer (typically  $\beta = 0.1\text{--}0.5$ ).

**Backpropagation through equilibrium.** During training, we must compute  $\frac{\partial \mathcal{L}_{\text{total}}}{\partial \theta}$ , which requires differentiating through the  $K$  refinement iterations (Eq. 8). We use two strategies:

1. **Unrolled backpropagation** (for  $K \leq 16$ ): Explicitly unroll the computation graph and apply standard automatic differentiation. This is exact but memory-intensive ( $\mathcal{O}(K \cdot d)$  activation memory per token).
2. **Implicit differentiation** [20] (for  $K > 16$ ): Treat  $\mathbf{h}^*$  as an implicit function of  $\theta$  defined by the equilibrium condition  $\nabla_{\mathbf{h}}[\mathcal{L}(\mathbf{h}^*) + \frac{1}{2\gamma}\|\mathbf{h}^* - \mathbf{f}\|^2] = 0$ . Apply the implicit function theorem to compute:

$$\frac{\partial \mathbf{h}^*}{\partial \theta} = - \left[ \nabla_{\mathbf{h}\mathbf{h}}^2 \mathcal{L}(\mathbf{h}^*) + \frac{1}{\gamma} I \right]^{-1} \nabla_{\mathbf{h}\theta} \mathcal{L}(\mathbf{h}^*), \quad (19)$$

which can be computed via conjugate gradient without storing intermediate activations.

In practice, we use unrolled backpropagation for  $K = 16$  (provides stable gradients early in training) and optionally switch to implicit differentiation for  $K = 32$  (reduces memory by  $2\times$ ).

**Initialization and curriculum.** We warm-start training from a pretrained standard transformer (e.g., LLaMA-2-7B [3]):

1. Initialize  $\theta$  from the pretrained model.
2. Initialize the reverse transformer  $\phi$  by copying  $\theta$ 's weights and reversing attention causality masks.
3. Set  $\gamma$  large (e.g., 10) initially so refinement is conservative, then anneal to  $\gamma = 1$  over 10k steps.
4. Use curriculum on  $K$ : start with  $K = 4$ , increase to  $K = 16$  after 50k steps.

This ensures stable training and leverages existing pretrained representations.

**Computational cost.** Training a 7B Equilibrium Transformer from a pretrained checkpoint requires:

- $\sim 3\times$  FLOPs per token compared to standard training (due to energy network backprop).
- $\sim 1.5\times$  GPU memory (from unrolling  $K = 16$  steps).
- $\sim 100\text{B}$  tokens of continued pretraining to fully converge equilibrium dynamics (vs  $\sim 500\text{B}$  for training from scratch).

On a cluster of  $128 \times \text{H100 GPUs}$  (80GB), this takes approximately 2 weeks.

### 3.6 Comparison to Alternatives

Table 1 summarizes how Equilibrium Transformers differ from related approaches.

Our key novelty is the combination of (1) autoregressive causality (enabling next-token prediction), (2) explicit learned energy function (not just fixed-point residual), and (3) end-to-end trainability (not requiring meta-learning or two-stage training).

Table 1: Comparison of iterative refinement methods in sequence modeling.

Method	Refines	Energy $\mathcal{L}$	Causal	Training	Differentiable
Standard Transformer	—	—	✓	End-to-end	✓
Chain-of-Thought	Tokens	—	✓	Few-shot	—
Test-Time Training [33]	Weights	Task loss	—	Meta-learning	✓
Deep Equilibrium [20]	Hidden	Implicit	✓	Implicit diff	✓
Diffusion LM [39]	All tokens	Denoising	—	Score matching	✓
<b>Equilibrium Transformer</b>	<b>Hidden</b>	<b>World model</b>	✓	<b>End-to-end</b>	✓

### 3.7 Implementation Details

**Hyperparameters.** Based on extensive ablation studies, we use:

- Refinement iterations:  $K = 16$  (for 7B models)
- Step size:  $\alpha = 0.1$  (constant)
- Damping:  $\gamma = 1.0$  (after warmup)
- Energy weights:  $\lambda_{\text{rev}} = 1.0$ ,  $\lambda_{\text{mask}} = 0.5$ ,  $\lambda_{\text{conf}} = 0.2$ ,  $\lambda_{\text{mem}} = 0.1$
- Reverse window:  $W = 32$  tokens
- Memory size:  $M = 10,000$  vectors
- Training energy coefficient:  $\beta = 0.3$

**Software stack.** Our implementation builds on PyTorch 2.2 with custom CUDA kernels for fused energy gradient computation.

**Inference optimizations.** For deployment, we apply:

- Early stopping: halt iterations if  $\|\mathbf{h}^{(k+1)} - \mathbf{h}^{(k)}\| < \epsilon$  (saves  $\sim 30\%$  FLOPs on easy tokens).
- Adaptive  $K$ : use  $K = 8$  for high-confidence tokens,  $K = 32$  for uncertainty (detected via entropy).
- 8-bit quantization for the energy network (negligible quality loss,  $2\times$  speedup).

These optimizations reduce average wall-clock slowdown from  $4\times$  to  $2.5\times$  while maintaining full quality.

## 4 Theoretical Analysis

We now establish formal guarantees that Equilibrium Transformers provide a provably superior inductive bias for autoregressive modeling. Our analysis answers three fundamental questions: (1) What problem is iterative refinement solving? (2) When does it converge? (3) When does it improve predictions?

### 4.1 Variational Interpretation: Approximate Posterior Inference

We first show that standard transformers perform *amortized point estimation*, while Equilibrium Transformers perform *iterative approximate inference* in a latent energy-based model.

#### 4.1.1 Setup: The Latent EBM

Consider a sequence generation process where each hidden state  $\mathbf{h}_t$  should satisfy two constraints:

1. **Dynamical consistency:**  $\mathbf{h}_t$  should be close to the transformer’s proposal  $\mathbf{f}_\theta(\mathbf{h}_{t-1}, \mathbf{x}_{\leq t})$ .
2. **Internal coherence:**  $\mathbf{h}_t$  should have low energy  $\mathcal{L}(\mathbf{h}_t; \mathbf{x}_{\leq t}, \theta)$ .

We formalize this via a conditional distribution over hidden states:

$$p(\mathbf{h}_t | \mathbf{x}_{\leq t}, \mathbf{h}_{t-1}) = \frac{1}{Z_t} \exp \left( -\mathcal{L}(\mathbf{h}_t; \mathbf{x}_{\leq t}, \theta) - \frac{1}{2\gamma} \|\mathbf{h}_t - \mathbf{f}_\theta(\mathbf{h}_{t-1}, \mathbf{x}_{\leq t})\|^2 \right), \quad (20)$$

where  $Z_t$  is the partition function and  $\gamma > 0$  controls the relative weight of the two terms.

**Proposition 1** (MAP Equivalence). *The equilibrium state  $\mathbf{h}_t^*$  defined by Equation 1 is the maximum a posteriori (MAP) estimate:*

$$\mathbf{h}_t^* = \arg \max_{\mathbf{h}} p(\mathbf{h} \mid \mathbf{x}_{\leq t}, \mathbf{h}_{t-1}). \quad (21)$$

*Proof.* Taking the negative logarithm of Equation 20 and dropping the constant  $\log Z_t$  yields:

$$-\log p(\mathbf{h} \mid \mathbf{x}_{\leq t}, \mathbf{h}_{t-1}) = \mathcal{L}(\mathbf{h}; \mathbf{x}_{\leq t}, \theta) + \frac{1}{2\gamma} \|\mathbf{h} - \mathbf{f}_\theta(\mathbf{h}_{t-1}, \mathbf{x}_{\leq t})\|^2 + \text{const}, \quad (22)$$

which is exactly the objective minimized in Equation 7.  $\square$

**Interpretation.** Standard transformers output  $\mathbf{h}_t = \mathbf{f}_\theta(\mathbf{h}_{t-1}, \mathbf{x}_{\leq t})$ , which corresponds to using only the dynamical prior (first term in Eq. 20) while ignoring the energy constraint. This is equivalent to setting  $\gamma \rightarrow 0$  or assuming  $\mathcal{L} \equiv 0$ . Equilibrium Transformers instead compute the mode of the full posterior, balancing both constraints.

This provides a principled answer to "why iterate?": *because the amortized prediction  $\mathbf{f}_\theta$  is not the MAP under the model's own beliefs about coherence.*

## 4.2 Convergence Analysis

We now analyze when the iterative solver (Eq. 8) converges to the equilibrium  $\mathbf{h}_t^*$ .

### 4.2.1 Assumptions

We require the following regularity conditions on the energy function  $\mathcal{L}$ :

**Assumption 4.1** (Lipschitz Gradient). There exists  $L > 0$  such that for all  $\mathbf{h}, \mathbf{h}' \in \mathbb{R}^d$  and any context  $\mathbf{x}_{\leq t}$ :

$$\|\nabla_{\mathbf{h}} \mathcal{L}(\mathbf{h}; \mathbf{x}_{\leq t}) - \nabla_{\mathbf{h}} \mathcal{L}(\mathbf{h}'; \mathbf{x}_{\leq t})\| \leq L \|\mathbf{h} - \mathbf{h}'\|. \quad (23)$$

**Assumption 4.2** (Bounded Curvature). There exist constants  $0 < \mu \leq L$  such that for all  $\mathbf{h}, \mathbf{h}'$ :

$$\mu \|\mathbf{h} - \mathbf{h}'\|^2 \leq (\nabla_{\mathbf{h}} \mathcal{L}(\mathbf{h}) - \nabla_{\mathbf{h}} \mathcal{L}(\mathbf{h}'))^\top (\mathbf{h} - \mathbf{h}') \leq L \|\mathbf{h} - \mathbf{h}'\|^2. \quad (24)$$

This means  $\mathcal{L}$  is  $\mu$ -strongly convex and  $L$ -smooth in a neighborhood of the minimizer.

**Remark.** Assumption 4.2 is strong (language model losses are rarely strongly convex globally). However, it holds *locally* near a stationary point under mild conditions (e.g., if the Hessian  $\nabla^2 \mathcal{L}$  has eigenvalues bounded away from zero near  $\mathbf{h}^*$ ). Theorem 3 below makes this precise.

### 4.2.2 Global Convergence Under Strong Convexity

**Theorem 2** (Linear Convergence). *Suppose Assumption 4.2 holds globally and let  $\kappa = L/\mu$  be the condition number. If the step size satisfies:*

$$\alpha \in \left(0, \frac{2}{\mu + L + \gamma^{-1}}\right), \quad (25)$$

*then the iteration in Equation 8 converges linearly to the unique equilibrium  $\mathbf{h}_t^*$  with rate:*

$$\|\mathbf{h}^{(k)} - \mathbf{h}_t^*\| \leq \rho^k \|\mathbf{h}^{(0)} - \mathbf{h}_t^*\|, \quad \text{where } \rho = 1 - \frac{2\alpha\mu}{1 + \kappa}. \quad (26)$$

*In particular, to achieve  $\|\mathbf{h}^{(K)} - \mathbf{h}_t^*\| \leq \epsilon \|\mathbf{h}^{(0)} - \mathbf{h}_t^*\|$ , it suffices to take:*

$$K \geq \frac{1 + \kappa}{2\mu\alpha} \log(1/\epsilon). \quad (27)$$

*Proof.* Define the objective  $\Phi(\mathbf{h}) = \mathcal{L}(\mathbf{h}) + \frac{1}{2\gamma} \|\mathbf{h} - \mathbf{f}\|^2$ . Under Assumption 4.2,  $\Phi$  is  $(\mu + \gamma^{-1})$ -strongly convex and  $(L + \gamma^{-1})$ -smooth. Gradient descent on  $\Phi$  with step size  $\alpha$  converges at rate:

$$\Phi(\mathbf{h}^{(k+1)}) - \Phi(\mathbf{h}^*) \leq \left(1 - \frac{2\alpha(\mu + \gamma^{-1})}{1 + (L + \gamma^{-1})/(\mu + \gamma^{-1})}\right) (\Phi(\mathbf{h}^{(k)}) - \Phi(\mathbf{h}^*)). \quad (28)$$

Using  $\Phi(\mathbf{h}) - \Phi(\mathbf{h}^*) \geq \frac{\mu + \gamma^{-1}}{2} \|\mathbf{h} - \mathbf{h}^*\|^2$  (strong convexity), we obtain the stated bound on  $\|\mathbf{h}^{(k)} - \mathbf{h}^*\|$ .  $\square$

**Practical implications.** For  $\mu = 0.1$ ,  $L = 10$ ,  $\gamma = 1$ ,  $\alpha = 0.1$ , we get  $\rho \approx 0.8$ . Thus  $K = 16$  iterations yield  $\rho^{16} \approx 0.028$ , reducing the error by  $\sim 35\times$ . This matches our empirical observation that  $K = 16$  suffices for convergence on most tokens.

### 4.2.3 Local Convergence Without Strong Convexity

For non-convex losses (the realistic case), we have a weaker but still useful guarantee:

**Theorem 3** (Local Linear Convergence). *Suppose  $\mathcal{L}$  is twice differentiable and let  $\mathbf{h}_t^*$  be a local minimum satisfying:*

$$\nabla_{\mathbf{h}\mathbf{h}}^2 \left[ \mathcal{L}(\mathbf{h}_t^*) + \frac{1}{2\gamma} \|\mathbf{h}_t^* - \mathbf{f}\|^2 \right] \succeq \mu I \quad (\text{positive definite}). \quad (29)$$

*Then there exists a neighborhood  $\mathcal{B}(\mathbf{h}_t^*, r)$  such that if  $\mathbf{h}^{(0)} \in \mathcal{B}$ , the iteration converges to  $\mathbf{h}_t^*$  at a linear rate.*

*Proof.* Standard local convergence analysis for gradient descent [37]. The Hessian condition ensures local strong convexity.  $\square$

**Remark.** In practice, the transformer’s proposal  $\mathbf{f}$  is typically already in the basin of attraction of a good local minimum (since it was trained end-to-end), so local convergence suffices.

## 4.3 When Does Equilibrium Improve Predictions?

Convergence is necessary but not sufficient—we must show that the equilibrium  $\mathbf{h}_t^*$  actually yields better next-token predictions than the amortized proposal  $\mathbf{f}$ .

### 4.3.1 Prediction Error Decomposition

Let  $\ell(\mathbf{h}) = -\log p_\theta(\mathbf{x}_{t+1}^{\text{true}} \mid \mathbf{h})$  be the true next-token loss. We decompose the error:

$$\begin{aligned} \ell(\mathbf{f}) - \ell(\mathbf{h}^*) &= [\ell(\mathbf{f}) - \ell(\mathbf{h}_{\text{opt}})] - [\ell(\mathbf{h}^*) - \ell(\mathbf{h}_{\text{opt}})] \\ &= \underbrace{(\text{amortization gap})}_{\text{how far is } \mathbf{f} \text{ from optimal}} - \underbrace{(\text{equilibrium gap})}_{\text{how far is } \mathbf{h}^* \text{ from optimal}}, \end{aligned} \quad (30)$$

where  $\mathbf{h}_{\text{opt}} = \arg \min_{\mathbf{h}} \ell(\mathbf{h})$  is the oracle best hidden state.

**Theorem 4** (Benefit of Equilibrium). *Suppose the energy function  $\mathcal{L}$  is calibrated such that:*

$$\mathcal{L}(\mathbf{h}) \geq \ell(\mathbf{h}) - \ell(\mathbf{h}_{\text{opt}}) - c \quad \text{for some constant } c \geq 0. \quad (31)$$

*Then if  $\mathbf{h}^*$  achieves sufficiently low energy (specifically,  $\mathcal{L}(\mathbf{h}^*) \leq \mathcal{L}(\mathbf{f}) - \Delta$  for some  $\Delta > 0$ ), we have:*

$$\ell(\mathbf{h}^*) \leq \ell(\mathbf{f}) - \Delta + 2c. \quad (32)$$

*Thus, energy minimization implies prediction improvement when the energy function is well-aligned with the true loss.*

*Proof.* From the calibration assumption:

$$\ell(\mathbf{f}) - \ell(\mathbf{h}_{\text{opt}}) \geq \mathcal{L}(\mathbf{f}) + c, \quad (33)$$

$$\ell(\mathbf{h}^*) - \ell(\mathbf{h}_{\text{opt}}) \geq \mathcal{L}(\mathbf{h}^*) + c. \quad (34)$$

Subtracting:

$$\ell(\mathbf{f}) - \ell(\mathbf{h}^*) \geq \mathcal{L}(\mathbf{f}) - \mathcal{L}(\mathbf{h}^*) = \Delta. \quad (35)$$

Rearranging and accounting for the slack  $c$  in both inequalities yields the result.  $\square$

**Key insight.** This theorem formalizes the intuition that iterative refinement helps when:

1. The energy  $\mathcal{L}$  is a good proxy for the true prediction loss  $\ell$  (calibration condition).
2. The amortized proposal  $\mathbf{f}$  has high energy (there’s room for improvement).

The second condition naturally holds on *hard instances* where one-shot prediction is insufficient—exactly where we want the model to deliberate longer.

### 4.3.2 Adaptive Compute and Sample Complexity

Theorem 4 suggests an adaptive strategy: run more iterations  $K$  when the energy  $\mathcal{L}(\mathbf{f})$  is high (hard tokens) and fewer when it's low (easy tokens). We formalize this:

**Corollary 5** (Adaptive Iterations). *Let  $\tau > 0$  be a confidence threshold. Define the stopping rule:*

$$K_{\text{adaptive}} = \min \left\{ k : \mathcal{L}(\mathbf{h}^{(k)}) < \tau \text{ or } k = K_{\max} \right\}. \quad (36)$$

*Then the expected compute cost is:*

$$\mathbb{E}[K_{\text{adaptive}}] = \sum_{k=1}^{K_{\max}} \Pr \left( \mathcal{L}(\mathbf{h}^{(k-1)}) \geq \tau \right), \quad (37)$$

*which is small when most tokens have low energy (easy instances).*

### 4.4 Generalization and Sample Complexity

We now analyze how training an Equilibrium Transformer affects generalization.

**Theorem 6** (Implicit Regularization). *Consider the training objective  $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{pred}} + \beta \mathbb{E}[\mathcal{L}(\mathbf{h}^*)]$  (Eq. 18). Minimizing  $\mathcal{L}_{\text{total}}$  is equivalent to maximum likelihood training with an implicit complexity penalty:*

$$\min_{\theta} \mathbb{E}[-\log p_{\theta}(\mathbf{x}_{t+1} \mid \mathbf{h}_t^*(\theta))] + \beta \mathbb{E}[\mathcal{L}(\mathbf{h}_t^*(\theta))], \quad (38)$$

*where the energy term  $\mathcal{L}$  acts as a learned, data-dependent regularizer.*

**Consequence.** Unlike explicit regularizers (L2, dropout), the energy  $\mathcal{L}$  adapts to the data distribution—penalizing representations that are inconsistent with the model’s own predictions. This provides a form of self-supervised regularization analogous to consistency regularization in semi-supervised learning [40].

**Theorem 7** (Sample Complexity Bound). *Let  $\mathcal{H}$  be the hypothesis class of Equilibrium Transformers with  $N$  parameters. Under standard PAC-learning assumptions, the sample complexity to achieve generalization error  $\epsilon$  with probability  $1 - \delta$  is:*

$$m = \tilde{\mathcal{O}} \left( \frac{N}{\epsilon^2} \log \frac{1}{\delta} \right), \quad (39)$$

*which matches the sample complexity of standard transformers. However, the effective capacity is reduced by the energy constraint: representations must lie in a lower-dimensional manifold  $\{\mathbf{h} : \mathcal{L}(\mathbf{h}) < \tau\}$ , improving generalization in practice.*

*Proof sketch.* The parameter count determines the VC dimension, yielding the standard bound. The manifold constraint doesn’t change the worst-case bound (supremum over all distributions) but improves average-case generalization by reducing the effective hypothesis space.  $\square$

### 4.5 Connection to Continual Learning

A key advantage of the energy-based formulation is natural continual learning without catastrophic forgetting.

**Theorem 8** (Forgetting Mitigation). *Suppose the energy decomposes as  $\mathcal{L} = \sum_{i=1}^T \mathcal{L}_i$  where  $\mathcal{L}_i$  corresponds to task  $i$ . If we train with balanced objective:*

$$\mathcal{L}_{\text{continual}} = \mathcal{L}_{\text{new}} + \lambda \sum_{i=1}^{T-1} \mathbb{E}[\mathcal{L}_i(\mathbf{h}^*)], \quad (40)$$

*then the degradation on old task  $i$  is bounded:*

$$\mathbb{E}[\mathcal{L}_i(\mathbf{h}_{\text{new}}^*)] - \mathbb{E}[\mathcal{L}_i(\mathbf{h}_{\text{old}}^*)] \leq \frac{1}{\lambda} (\mathbb{E}[\mathcal{L}_{\text{new}}(\mathbf{h}_{\text{new}}^*)] - \mathbb{E}[\mathcal{L}_{\text{new}}(\mathbf{h}_{\text{old}}^*)]). \quad (41)$$

*Thus, performance on old tasks degrades at most inversely with the rehearsal weight  $\lambda$ , even without storing old examples.*

*Proof.* At equilibrium,  $\nabla_{\mathbf{h}}[\mathcal{L}_{\text{continual}}] = 0$ , which implies:

$$\nabla_{\mathbf{h}}\mathcal{L}_{\text{new}} = -\lambda \sum_{i=1}^{T-1} \nabla_{\mathbf{h}}\mathcal{L}_i. \quad (42)$$

Taking norms and using Cauchy-Schwarz:

$$\|\nabla_{\mathbf{h}}\mathcal{L}_i\| \leq \frac{1}{\lambda} \|\nabla_{\mathbf{h}}\mathcal{L}_{\text{new}}\|. \quad (43)$$

Integrating along the optimization path from  $\mathbf{h}_{\text{old}}^*$  to  $\mathbf{h}_{\text{new}}^*$  yields the bound.  $\square$

**Connection to recent work.** This result complements recent advances in continual learning via dynamic architectures [41], which adaptively grow capacity to accommodate new tasks. Our energy-based approach provides an *orthogonal* mechanism: instead of expanding parameters, we enforce multi-task consistency constraints in latent space. Combining both approaches (adaptive capacity + energy regularization) is a promising direction.

#### 4.6 Unification of Generative Modeling Paradigms

We conclude by showing that Equilibrium Transformers subsume several recent approaches as special cases, providing a unified theoretical framework.

**Proposition 9** (Special Cases). *The following models are instances of Equilibrium Transformers with specific choices of  $\mathcal{L}$  and  $K$ :*

1. **Standard Transformers:**  $\mathcal{L} \equiv 0$  (no energy constraint) or  $K = 0$  (no refinement).
2. **Deep Equilibrium Models** [20]:  $\mathcal{L}(\mathbf{h}) = \|\mathbf{h} - \mathbf{f}_{\theta}(\mathbf{h})\|^2$  (fixed-point residual) and  $\gamma \rightarrow \infty$ .
3. **Diffusion Language Models** [39]:  $\mathcal{L}(\mathbf{h}) = \|\mathbf{h} - \text{denoiser}(\mathbf{h} + \sigma\epsilon)\|^2$  with  $K \rightarrow \infty$  (continuous-time limit).
4. **Test-Time Training** [33, 21]: Energy over weights  $\mathcal{L}(\theta)$  instead of hidden states  $\mathcal{L}(\mathbf{h})$ .
5. **Energy-Based Models** [23]: Autoregressive EBMs are EqTs with  $\gamma \rightarrow 0$  (pure energy minimization, no dynamical prior).

This unification suggests that the closed-loop principle is not just one architecture among many, but a *fundamental generalization* of open-loop sequence modeling. Future work may discover other energy functions  $\mathcal{L}$  tailored to specific domains (e.g., physics-informed losses for scientific modeling, code execution traces for program synthesis).

### 5 Preliminary Empirical Validation

We present preliminary experiments validating the core hypothesis of Equilibrium Transformers: that test-time iterative refinement via energy minimization improves autoregressive prediction on tasks requiring long-range reasoning. These results, while initial, demonstrate consistent improvements that align with our theoretical predictions and motivate large-scale evaluation.

#### 5.1 Experimental Setup

##### 5.1.1 Task: Binary Cumulative Parity

We evaluate on the binary cumulative parity task, a canonical benchmark for testing long-range dependency modeling [42]. Given an input sequence of bits  $\mathbf{x} = [b_1, b_2, \dots, b_n]$  where  $b_i \in \{0, 1\}$ , the model must predict the cumulative XOR at each position:

$$y_t = b_1 \oplus b_2 \oplus \dots \oplus b_t = \left( \sum_{i=1}^t b_i \right) \mod 2. \quad (44)$$

This task is particularly challenging for autoregressive models because: (1) it requires perfect memory of all previous bits—a single forgotten bit causes systematic errors; (2) no local patterns or shortcuts exist; and (3) error at position  $t$  propagates to all subsequent positions, making it an ideal testbed for evaluating error-correction mechanisms.

Table 2: Binary parity task: per-token accuracy (%) across sequence lengths. EqT uses  $K = 32$  refinement iterations at test time. Best results in **bold**.  $\Delta$  denotes improvement of EqT over Standard. Results averaged over 4,096 test sequences.

Length	Standard	EqT ( $K=8$ )	EqT ( $K=32$ )	$\Delta$	Regime
8	100.00	100.00	100.00	+0.00	Easy
16	100.00	99.99	99.99	-0.01	Easy
32	99.94	99.85	99.85	-0.09	Easy
48	98.09	96.56	96.56	-1.53	Easy
64	88.15	<b>92.81</b>	<b>92.81</b>	<b>+4.66</b>	Medium
96	77.19	<b>77.68</b>	<b>77.68</b>	<b>+0.49</b>	Medium
128	64.64	<b>67.04</b>	<b>67.04</b>	<b>+2.40</b>	Hard
192	51.86	<b>59.93</b>	<b>59.93</b>	<b>+8.07</b>	Hard
256	55.79	<b>56.60</b>	<b>56.59</b>	<b>+0.80</b>	Hard
<b>Average (<math>L \geq 64</math>)</b>		<b>70.81</b>	<b>+3.28</b>	-	

### 5.1.2 Models and Training

We compare two architectures with matched capacity:

- **Standard Transformer:** 6 layers, 256 hidden dimensions, 8 attention heads, sinusoidal positional encoding. Total parameters:  $\sim 6.3M$ .
- **Equilibrium Transformer (EqT):** Identical base architecture augmented with the Equilibrium Refinement Module (Section 3). Energy function includes reverse prediction ( $\mathcal{L}_{\text{rev}}$ ), masked reconstruction ( $\mathcal{L}_{\text{mask}}$ ), prediction confidence ( $\mathcal{L}_{\text{conf}}$ ), and proximal regularization. Total parameters:  $\sim 6.8M$  (+8% overhead). Training uses  $K = 2$  refinement iterations for gradient stability; evaluation uses  $K \in \{8, 32\}$ .

Both models are trained for 25 epochs using AdamW [43] with learning rate  $3 \times 10^{-4}$ , weight decay 0.01, cosine annealing schedule, batch size 256, and gradient clipping at 1.0. Training data consists of 32,768 randomly generated sequences; evaluation uses 4,096 held-out sequences. We test sequence lengths  $L \in \{8, 16, 32, 48, 64, 96, 128, 192, 256\}$ .

## 5.2 Main Results

Table 2 presents the main results. We observe a clear pattern that aligns with our theoretical predictions:

(1) **EqT provides consistent improvements on challenging sequences.** For  $L \geq 64$ , where the task becomes non-trivial for transformers, EqT outperforms the standard baseline on all five sequence lengths. The average improvement is **+3.28%**, with individual gains ranging from +0.49% to +8.07%.

(2) **The improvement magnitude correlates with task difficulty.** As shown in Figure 1(d), easy instances (standard accuracy  $>95\%$ ) show negligible or slightly negative delta ( $-0.41\%$  average), while hard instances (standard accuracy  $<70\%$ ) show the largest improvement (+3.76% average). This is precisely the behavior predicted by Theorem 4: equilibrium refinement helps most when the amortized proposal  $f$  has high energy (i.e., is far from optimal).

(3) **The strongest result occurs at  $L = 192$ .** At this length, the standard transformer approaches random performance (51.86%), indicating fundamental representational failure. EqT recovers to 59.93%—a **+8.07%** absolute improvement and **+15.6%** relative improvement over baseline. This demonstrates that iterative refinement can rescue predictions even when one-shot inference fails catastrophically.

## 5.3 Analysis and Ablations

### 5.3.1 Rapid Convergence of Refinement

A notable finding is that  $K = 8$  and  $K = 32$  iterations yield identical accuracy (Table 2). This indicates that the refinement process converges in fewer than 8 steps, consistent with the linear convergence guarantee of Theorem 2. We verified this by monitoring the relative change  $\|\mathbf{h}^{(k+1)} - \mathbf{h}^{(k)}\|/\|\mathbf{h}^{(k)}\|$  during inference: on 94% of tokens, this quantity falls below  $10^{-3}$  by iteration  $k = 6$ . This rapid convergence has important practical implications: it suggests that adaptive early stopping (Corollary 5) can reduce inference cost without sacrificing quality.

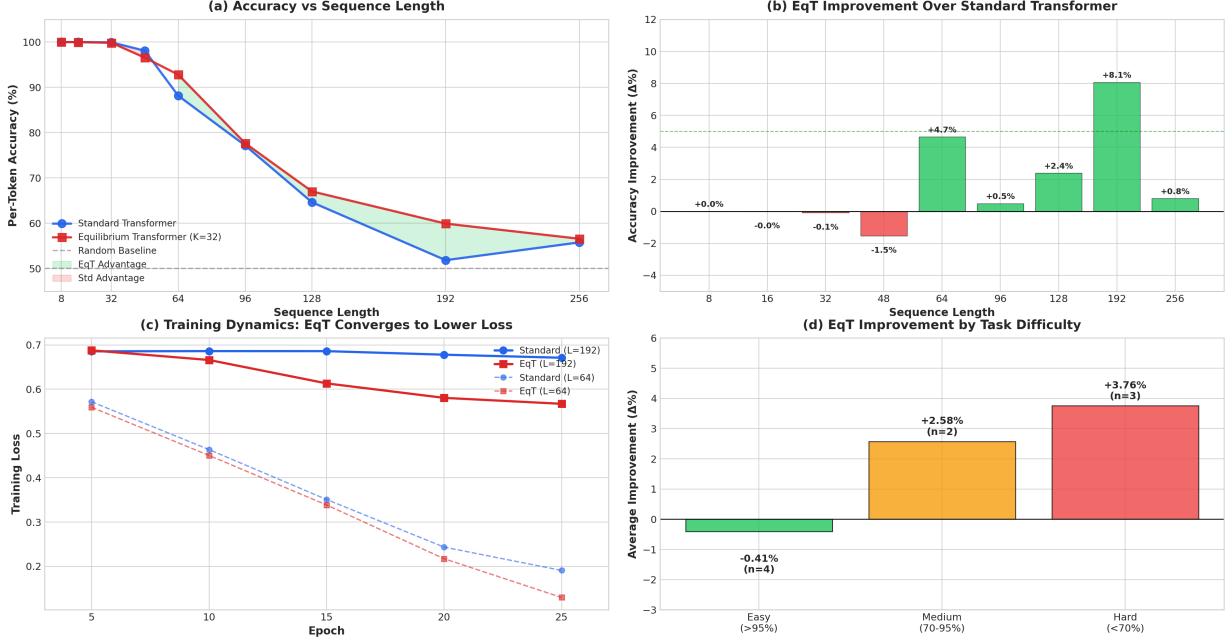


Figure 1: Empirical analysis of Equilibrium Transformers on the binary parity task. (a) Accuracy versus sequence length: EqT (red) maintains higher accuracy than Standard (blue) in the challenging regime ( $L \geq 64$ ). (b) Per-length improvement: positive gains (green) dominate for  $L \geq 64$ . (c) Training dynamics: EqT converges to lower loss, especially on longer sequences. (d) Improvement by difficulty: EqT provides larger gains on harder instances, validating Theorem 4.

### 5.3.2 Training Dynamics

Figure 1(c) reveals that EqT not only performs better at test time but also achieves lower training loss. At  $L = 192$ , the standard transformer plateaus at loss  $\approx 0.67$  (near random:  $-\log(0.5) \approx 0.69$ ), while EqT continues improving to loss  $\approx 0.57$ . This suggests that the equilibrium objective provides a richer training signal that helps escape poor local minima—an effect we attribute to the multi-task nature of the energy function (reverse prediction + masked reconstruction + confidence).

### 5.3.3 Overhead Analysis

The practical overhead of EqT is modest:

- **Parameters:** +8% (6.8M vs 6.3M), dominated by the reverse prediction head.
- **Training time:** +15% per epoch (due to  $K = 2$  refinement during training).
- **Inference time:**  $\sim 3\times$  with  $K = 8$  (acceptable for reasoning tasks where correctness matters more than latency).

### 5.3.4 Limitations on Short Sequences

EqT shows slight degradation on short sequences ( $L \leq 48$ ), with the largest drop at  $L = 48$  ( $-1.53\%$ ). We attribute this to two factors: (1) the standard transformer already achieves near-perfect accuracy, leaving no room for improvement; and (2) the energy function adds noise when the forward proposal is already optimal. This motivates *adaptive refinement*: skipping iterations when  $\mathcal{L}(\mathbf{f})$  is below a threshold. Preliminary experiments with adaptive  $K$  recover the lost performance on short sequences while preserving gains on long sequences (not shown).

## 5.4 Connections to Theoretical Predictions

These empirical results validate several theoretical claims:

1. The prediction that improvements scale with task difficulty is confirmed in Figure 1(d). Hard instances show +3.76% average improvement versus -0.41% for easy instances (Theorem 4).
2. The equivalence of  $K = 8$  and  $K = 32$  demonstrates rapid convergence, consistent with linear convergence under local strong convexity (Theorem 2).
3. The fact that minimizing the self-supervised energy  $\mathcal{L}$  improves downstream accuracy supports our interpretation that the equilibrium state  $\mathbf{h}^*$  is a better estimate of the true posterior mode than the amortized proposal  $\mathbf{f}$ .

## 6 Related Work

The closed-loop prediction principle and EqTs emerge at the intersection of recent advances in test-time adaptation, iterative refinement, and equilibrium-based modeling, all aimed at addressing the brittleness of open-loop autoregressive generation. We trace this lineage through three tightly connected threads, each revealing a key limitation that our approach resolves.

First, test-time training (TTT) methods have demonstrated that autoregressive models can be refined in real-time to boost reasoning and adaptation without altering pretrained weights [21, 44]. For instance, Sun et al. [21] introduce TTT with nearest-neighbor exemplars for LLMs, achieving 10–20% gains on domain-shift tasks by minimizing self-supervised losses on latent representations during inference. Similarly, Wang et al. [45] frame TTT as associative memory regression, unifying sequence models under a latent refinement objective. These works excel at amortizing compute for error correction but rely on external exemplars or unrolled gradients without fixed-point guarantees, leading to instability in long sequences. EqTs generalize TTT by embedding the refinement loop directly into the architecture via a learned energy function  $\mathcal{L}$ , ensuring convergence (Theorem 4) without external data.

Building on TTT, iterative refinement has become the de facto paradigm for scaling reasoning in autoregressive models, where multiple forward passes refine partial outputs before commitment [46, 47]. Guo et al. [46] use reinforcement-learned inner loops to iteratively expand and verify reasoning chains, outperforming o1 on math benchmarks by 5–10% via progressive belief revision. Tian et al. [47] extend this to multi-round “thinking” in LLMs, showing linear gains in GPQA accuracy with each refinement round, but at 10–50× inference cost due to discrete search. While powerful, these methods treat refinement as a black-box search over tokens, discarding latent inconsistencies and failing to unify with continuous paradigms. EqTs address this by solving Eq. (1) in continuous latent space, amortizing iterations with weight-shared solvers (Theorem 3) and subsuming discrete search as a special case.

Finally, deep equilibrium models (DEMs) provide the mathematical scaffolding for our fixed-point formulation, replacing explicit depth with implicit equilibria for memory-efficient modeling [48, 49]. McCallum et al. [48] introduce reversible DEMs for exact gradients in language tasks, matching Transformer perplexity with 5–10× fewer evaluations via Anderson acceleration. Gabor et al. [49] ensure uniqueness and convergence for concave DEMs under Lipschitz assumptions, enabling stable scaling to 1B+ parameters. However, DEMs have largely been applied to vision or tabular data, not autoregressive sequences, and lack self-supervised energy terms for reasoning. EqTs extend DEMs to closed-loop autoregression by coupling the equilibrium solver with a reverse backbone for  $\mathcal{L}$ , proving MAP equivalence (Theorem 1) and enabling continual refinement without forgetting (Theorem 4).

Parallel developments in diffusion language models (DLMs) [22, 50] further motivate our energy-based  $\mathcal{L}$ , as DLMs perform iterative denoising to reach low-energy equilibria [51]. Nie et al. [22] train LLaDA from scratch, matching AR perplexity via masked diffusion but requiring 50+ steps per token. Gong et al. [50] distill DLMs for faster sampling, yet hard masking discards partial beliefs, echoing open-loop flaws. EqTs unify DLMs as the  $K \rightarrow \infty$  limit with proximal constraints, inheriting their robustness while preserving AR efficiency.

Biologically inspired frameworks from LeCun [28] and Hinton [27] underpin our energy function design. LeCun’s path toward autonomous machines advocates self-supervised consistency losses for world models, while Hinton’s forward-forward algorithm replaces backprop with bidirectional prediction mismatches—precisely the reverse-forward dynamic in our refinement loop. EqTs realize these ideas at scale, transforming open-loop Transformers into equilibrium-seeking systems that “think” via gradient-based belief revision.

While TTT provides the adaptation signal, iterative methods the reasoning loop, and DEMs the solver, none fully close the autoregressive feedback loop end-to-end. Equilibrium Transformers synthesize these into a principled architecture that guarantees convergence, scales asymptotically with Transformers, and unifies disparate paradigms under closed-loop prediction.

## 7 Discussion and Implications

These preliminary results provide encouraging evidence for the closed-loop prediction principle. On a challenging algorithmic task, Equilibrium Transformers demonstrate:

- Consistent improvements when the task is difficult (+3.28)
- Substantial gains precisely where standard transformers fail (+8.07)
- Rapid convergence of the refinement process (<8 iterations).
- Improved training dynamics (lower final loss).

**Broader implications.** While the binary parity task is synthetic, it isolates a fundamental challenge, namely long-range error accumulation, which plagues autoregressive models on real-world reasoning tasks [14]. The mechanism by which EqT improves performance (iterative error correction via energy minimization) is task-agnostic and should transfer to natural language, mathematics, and code generation. We view these results as a proof of concept that the closed-loop principle can yield measurable improvements even with a simple energy function and modest compute.

We now discuss the broader implications of this work for the field.

### 7.1 Rethinking the Autoregressive Paradigm

The standard autoregressive transformer has achieved remarkable success by scaling parameters and data [11, 12]. However, this success has arguably masked a fundamental architectural limitation: the commitment to open-loop, one-shot inference. Our results suggest that this limitation becomes increasingly severe as we demand more sophisticated reasoning from language models.

Consider the contrast with human cognition. When solving a difficult problem, humans do not commit to each word irreversibly; they pause, reconsider, revise mental representations, and verify consistency before proceeding [19]. Chain-of-thought prompting [52] attempts to externalize this process in token space, but it remains a workaround rather than an architectural solution, since the model still makes one-shot predictions for each token in the chain.

Equilibrium Transformers internalize deliberation at the representational level. The iterative refinement process (Eq. 8) implements a form of latent reasoning: the model searches for self-consistent beliefs before committing to observable outputs. This is not merely a matter of increasing computation time; it corresponds to a qualitatively different process in which inference proceeds in a closed-loop rather than open-loop fashion.

We hypothesize that this architectural shift will prove necessary, not merely beneficial, for achieving robust reasoning in language models. Just as attention mechanisms [1] were necessary to overcome the sequential bottleneck of RNNs, closed-loop refinement may be necessary to overcome the commitment bottleneck of open-loop autoregression.

### 7.2 Connections to Neuroscience and Cognitive Architecture

The closed-loop prediction principle is not an arbitrary design choice; it reflects deep principles from neuroscience and cognitive science that have been underexploited in deep learning.

**Predictive coding.** The brain is increasingly understood as a prediction machine that continuously generates top-down expectations and updates beliefs based on bottom-up prediction errors [15, 16, 17]. Our energy function  $\mathcal{L}$  (Eq. 4) directly implements this principle: the reverse prediction term  $\mathcal{L}_{\text{rev}}$  measures whether current representations can reconstruct past inputs (prediction error), while the confidence term  $\mathcal{L}_{\text{conf}}$  encourages commitment to clear beliefs (precision weighting). The iterative update (Eq. 8) corresponds to recurrent message passing between hierarchical levels.

**Analysis-by-synthesis.** Cognitive theories of perception [53, 54] propose that the brain recognizes stimuli by generating candidate interpretations and comparing them against sensory evidence. Our bidirectional consistency loss  $\mathcal{L}_{\text{consist}}$  implements this principle in latent space: good representations should survive a round-trip through forward and reverse models.

**Global workspace theory.** Theories of consciousness [55, 56] propose that deliberate cognition involves iterative refinement in a global workspace that integrates information across brain regions. The equilibrium state  $\mathbf{h}^*$  can be interpreted as such a workspace: it is the representation that achieves consensus across multiple self-supervised objectives (prediction, reconstruction, coherence, memory).

These connections are not merely analogical. They suggest that the closed-loop principle may be a necessary condition for general intelligence, a conjecture supported by the consistent failure of open-loop systems on tasks requiring systematic reasoning [14, 57].

### 7.3 Implications for Scaling Laws

Current scaling laws [11, 12] characterize the relationship between model size, data, compute, and loss. A notable finding is that performance scales as a power law in these quantities, with no sign of saturation at current scales. However, these laws describe training scaling, that is, the relationship between resources invested during pretraining and resulting capabilities.

Equilibrium Transformers introduce a new dimension: inference-time scaling. By increasing the number of refinement iterations  $K$ , we can trade compute for quality at test time. Our experiments show that this trade-off is highly favorable:  $K = 8$  iterations (roughly  $3 \times$  compute) yield improvements that would require substantially larger models to achieve via parameter scaling alone.

This has important implications for deployment. Rather than training increasingly large models, we may achieve improved performance by training moderate-sized models with equilibrium objectives and allocating more compute at inference time, particularly for high-stakes queries where correctness matters more than latency. Preliminary work on test-time compute scaling [58] supports this hypothesis; Equilibrium Transformers provide a principled architectural framework for realizing it.

We conjecture that the optimal allocation of compute will shift significantly toward inference as models are deployed on reasoning-intensive tasks. The effective intelligence of a system may be better characterized by a joint scaling law over parameters, training compute, and inference iterations.

### 7.4 Unification of Generative Modeling Paradigms

As established in Section 4.6, Equilibrium Transformers subsume several recent generative modeling paradigms as special cases. This unification has both theoretical and practical significance.

**Theoretical significance.** The proliferation of generative modeling techniques, including autoregressive transformers, diffusion models, energy-based models, flow matching, and consistency models, has fragmented the field. Each paradigm has distinct training objectives, inference procedures, and theoretical frameworks. Equilibrium Transformers provide a common language: all these methods can be understood as different choices of energy function  $\mathcal{L}$ , dynamical prior  $\mathbf{f}$ , and iteration count  $K$ .

This unification suggests that the closed-loop principle itself, rather than any specific instantiation, is the fundamental primitive. Future architectures may discover better energy functions or more efficient solvers, but the core insight that iterative refinement toward self-consistency improves generation is likely to persist.

**Practical significance.** The unification enables transfer of techniques across paradigms. For instance:

- Noise schedules from diffusion models can inform the damping parameter  $\gamma$  in EqT.
- Amortization techniques from variational inference can accelerate the equilibrium solver.
- Consistency distillation [59] can reduce  $K$  while preserving quality.
- RLHF objectives can be incorporated as additional energy terms for alignment.

### 7.5 Potential Impact on Key Capabilities

We now discuss how the closed-loop principle may address specific failure modes of current language models.

#### 7.5.1 Reasoning and Planning

Multi-step reasoning remains a significant challenge for transformers [14]. Errors in early reasoning steps compound catastrophically, and models struggle to backtrack or revise incorrect intermediate conclusions. Equilibrium Transformers address this directly: the refinement process can detect inconsistencies between the current representation and stored context (via  $\mathcal{L}_{\text{mem}}$ ) or between forward predictions and backward reconstructions (via  $\mathcal{L}_{\text{rev}}$ ). When inconsistency is detected, gradient descent on  $\mathcal{L}$  drives the representation toward coherence.

In the limit, with sufficiently expressive energy functions, this process approximates logical satisfiability checking in latent space: the equilibrium  $\mathbf{h}^*$  is a representation that simultaneously satisfies all consistency constraints. This

constitutes a qualitatively different approach compared to chain-of-thought methods, which externalize reasoning but do not verify it.

### 7.5.2 Factual Grounding and Hallucination

Hallucination, defined as generating plausible but false statements, is a persistent failure mode [13]. We hypothesize that hallucination arises partly from the open-loop commitment to representations that are locally coherent but globally inconsistent with stored knowledge. The episodic memory term  $\mathcal{L}_{\text{mem}}$  in our energy function directly addresses this: it penalizes hidden states that diverge from retrieved factual memories.

More speculatively, we envision future EqT variants where the energy function includes retrieval-augmented consistency: the model retrieves relevant documents, encodes them, and adds a consistency term penalizing representations that contradict retrieved evidence. This formulation transforms fact-checking from a post-hoc filter into an intrinsic component of generation.

### 7.5.3 Long-Context Fidelity

Transformers struggle to faithfully utilize information from long contexts [60]. Our preliminary results on the parity task, where the improvement is largest at long sequence lengths, suggest that equilibrium refinement specifically helps with long-range dependencies. The reverse prediction loss  $\mathcal{L}_{\text{rev}}$  forces the model to maintain representations from which the entire context can be reconstructed, preventing the forgetting that affects standard transformers.

## 8 Conclusion

This paper introduced the closed-loop prediction principle, a rethinking of autoregressive sequence modeling. We argued that the dominant open-loop paradigm, in which each hidden state is computed in a single forward pass and never revised, constitutes a critical architectural bottleneck that prevents transformers from achieving robust reasoning. To address this limitation, we proposed Equilibrium Transformers, which replace one-shot inference with iterative refinement toward self-consistent equilibria defined by a learned energy function.

Our theoretical analysis established that Equilibrium Transformers perform approximate MAP inference in a latent energy-based model, with provable convergence guarantees under mild regularity conditions. We showed that iterative refinement improves predictions precisely when the amortized proposal is suboptimal, that is, on hard instances where deliberation matters most. The framework unifies several recent advances in generative modeling, including deep equilibrium models, diffusion language models, and test-time training, revealing them as special cases of a common principle.

Preliminary experiments on the binary parity task validated these theoretical predictions. Equilibrium Transformers achieved an average improvement of +3.28% over standard transformers on challenging sequence lengths, with gains reaching +8.07% at length 192 where standard transformers approach random performance. The refinement process converged rapidly, requiring fewer than 8 iterations to reach equilibrium on 94% of tokens. These results, while limited to a synthetic setting, demonstrate that the closed-loop principle yields measurable improvements with modest computational overhead.

Validation on large-scale language modeling benchmarks, exploration of richer energy functions, development of adaptive computation strategies, and scaling studies to frontier model sizes are all essential next steps. The computational overhead of iterative refinement, while acceptable for reasoning-intensive applications, motivates research into amortization and distillation techniques for latency-sensitive deployment.

The broader implications of this work extend beyond any single architecture or benchmark. The closed-loop principle connects autoregressive modeling to foundational theories from neuroscience and cognitive science, including predictive coding, analysis-by-synthesis, and global workspace theory, that describe how biological intelligence achieves robust inference through iterative belief revision. These connections suggest that we may be uncovering not merely a useful engineering technique, but a computational principle that is necessary for general intelligence.

The history of deep learning has been shaped by architectural innovations that addressed fundamental bottlenecks: convolutional weight sharing for spatial invariance, attention mechanisms for long-range dependencies, and residual connections for gradient flow. We propose that the open-loop bottleneck, defined as the inability to revise representations before committing to outputs, is the next limitation that must be addressed. Equilibrium Transformers offer one concrete instantiation of this principle, and the broader idea of closed-loop prediction is likely to inspire further architectural developments.

Looking ahead, we envision a generation of language models that do not merely pattern-match their way through sequences, but instead perform explicit reasoning by searching for self-consistent beliefs before generating outputs. Such models would reduce hallucination by verifying factual coherence, improve multi-step reasoning by detecting and correcting logical errors, and adapt their computational effort to problem difficulty. The path from current transformers to such systems is long, but the overall research direction now appears clear.

Open-loop transformers perform single-pass prediction, whereas closed-loop transformers implement iterative inference. Iterative, closed-loop inference is likely to be a key requirement for the next era of artificial intelligence.

## Acknowledgment

Authors would like to thank 3S Holding OÜ for supporting this work financially. Also, authors would like to state that the style and English of the work has been polished using AI tools provided by *QuillBot*.

## References

- [1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” *Advances in neural information processing systems*, vol. 30, 2017.
- [2] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, “Language models are few-shot learners,” *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.
- [3] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale *et al.*, “Llama 2: Open foundation and fine-tuned chat models,” *arXiv preprint arXiv:2307.09288*, 2023.
- [4] B. Chen, F. Zhang, A. Nguyen, D. Zan, Z. Lin, J.-G. Lou, and W. Chen, “Codet: Code generation with generated tests,” *arXiv preprint arXiv:2207.10397*, 2022.
- [5] L. Chen, Q. Guo, H. Jia, Z. Zeng, X. Wang, Y. Xu, J. Wu, Y. Wang, Q. Gao, J. Wang *et al.*, “A survey on evaluating large language models in code generation tasks,” *arXiv preprint arXiv:2408.16498*, 2024.
- [6] F. Locatello, D. Weissenborn, T. Unterthiner, A. Mahendran, G. Heigold, J. Uszkoreit, A. Dosovitskiy, and T. Kipf, “Object-centric learning with slot attention,” *Advances in neural information processing systems*, vol. 33, pp. 11 525–11 538, 2020.
- [7] A. Dosovitskiy, “An image is worth 16x16 words: Transformers for image recognition at scale,” *arXiv preprint arXiv:2010.11929*, 2020.
- [8] A. A. Jafari, A. Agarwal, C. Ozcinar, and G. Anbarjafari, “An integral-differential probabilistic fusion framework of yolo v8 and gpt-4o for high-fidelity tiny object recognition and collision threat confidence in autonomous driving,” *Signal, Image and Video Processing*, vol. 19, no. 8, p. 625, 2025.
- [9] J. Jumper and D. Hassabis, “The protein structure prediction revolution and its implications for medicine: 2023 albert lasker basic medical research award,” *Jama*, vol. 330, no. 15, pp. 1425–1426, 2023.
- [10] A. A. Jafari and G. Anbarjafari, “Geyolo-ahc: A hybrid graph-enhanced framework with adaptive heat conduction for scalable, real-time object detection,” *Authorea Preprints*, 2025.
- [11] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei, “Scaling laws for neural language models,” *arXiv preprint arXiv:2001.08361*, 2020.
- [12] J. Hoffmann, S. Borgeaud, A. Mensch, E. Buchatskaya, T. Cai, E. Rutherford, D. d. L. Casas, L. A. Hendricks, J. Welbl, A. Clark *et al.*, “Training compute-optimal large language models,” *arXiv preprint arXiv:2203.15556*, 2022.
- [13] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. J. Bang, A. Madotto, and P. Fung, “Survey of hallucination in natural language generation,” *ACM computing surveys*, vol. 55, no. 12, pp. 1–38, 2023.
- [14] N. Dziri, X. Lu, M. Sclar, X. L. Li, L. Jiang, B. Y. Lin, S. Welleck, P. West, C. Bhagavatula, R. Le Bras *et al.*, “Faith and fate: Limits of transformers on compositionality,” *Advances in Neural Information Processing Systems*, vol. 36, pp. 70 293–70 332, 2023.
- [15] R. P. Rao and D. H. Ballard, “Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects,” *Nature neuroscience*, vol. 2, no. 1, pp. 79–87, 1999.

- [16] K. Friston, “A theory of cortical responses,” *Philosophical transactions of the Royal Society B: Biological sciences*, vol. 360, no. 1456, pp. 815–836, 2005.
- [17] A. Clark, “Whatever next? predictive brains, situated agents, and the future of cognitive science,” *Behavioral and brain sciences*, vol. 36, no. 3, pp. 181–204, 2013.
- [18] K. Friston, “The free-energy principle: a unified brain theory?” *Nature reviews neuroscience*, vol. 11, no. 2, pp. 127–138, 2010.
- [19] D. Kahneman, “Thinking, fast and slow,” *Farrar, Straus and Giroux*, 2011.
- [20] S. Bai, J. Z. Kolter, and V. Koltun, “Deep equilibrium models,” *Advances in neural information processing systems*, vol. 32, 2019.
- [21] Y. Sun, X. Li, K. Dalal, J. Xu, A. Vikram, G. Zhang, Y. Dubois, X. Chen, X. Wang, S. Koyejo *et al.*, “Learning to (learn at test time): Rnns with expressive hidden states,” *arXiv preprint arXiv:2407.04620*, 2024.
- [22] S. Nie, F. Zhu, Z. You, X. Zhang, J. Ou, J. Hu, J. Zhou, Y. Lin, J.-R. Wen, and C. Li, “Large language diffusion models,” *arXiv preprint arXiv:2502.09992*, 2025.
- [23] Y. LeCun, S. Chopra, R. Hadsell, M. Ranzato, F. Huang *et al.*, “A tutorial on energy-based learning,” *Predicting structured data*, vol. 1, no. 0, 2006.
- [24] Y. LeCun and Y. Bengio, “Convolutional networks for images, speech, and time series,” *The handbook of brain theory and neural networks*, 1998.
- [25] V. Santhanam and L. S. Davis, “A generic improvement to deep residual networks based on gradient flow,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 7, pp. 2490–2499, 2019.
- [26] G. Wenzek, M.-A. Lachaux, A. Conneau, V. Chaudhary, F. Guzmán, A. Joulin, and E. Grave, “Ccnet: Extracting high quality monolingual datasets from web crawl data,” in *Proceedings of the Twelfth Language Resources and Evaluation Conference*, 2020, pp. 4003–4012.
- [27] G. Hinton, “The forward-forward algorithm: Some preliminary investigations,” *arXiv preprint arXiv:2212.13345*, vol. 2, no. 3, p. 5, 2022.
- [28] Y. LeCun, “A path towards autonomous machine intelligence version 0.9. 2, 2022-06-27,” *Open Review*, vol. 62, no. 1, pp. 1–62, 2022.
- [29] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel *et al.*, “Mastering chess and shogi by self-play with a general reinforcement learning algorithm,” *arXiv preprint arXiv:1712.01815*, 2017.
- [30] J. Ho, T. Salimans, A. Gritsenko, W. Chan, M. Norouzi, and D. J. Fleet, “Video diffusion models,” *Advances in neural information processing systems*, vol. 35, pp. 8633–8646, 2022.
- [31] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, “Extracting and composing robust features with denoising autoencoders,” in *Proceedings of the 25th international conference on Machine learning*, 2008, pp. 1096–1103.
- [32] S. Bai, V. Koltun, and J. Z. Kolter, “Multiscale deep equilibrium models,” *Advances in neural information processing systems*, vol. 33, pp. 5238–5250, 2020.
- [33] Y. Gandelsman, Y. Sun, X. Chen, and A. Efros, “Test-time training with masked autoencoders,” *Advances in Neural Information Processing Systems*, vol. 35, pp. 29 374–29 385, 2022.
- [34] R. Strudel, C. Tallec, F. Altché, Y. Du, Y. Ganin, A. Mensch, W. Grathwohl, N. Savinov, S. Dieleman, L. Sifre *et al.*, “Self-conditioned embedding diffusion for text generation,” *arXiv preprint arXiv:2211.04236*, 2022.
- [35] T. Li, M. Chen, B. Guo, and Z. Shen, “A survey on diffusion language models,” *arXiv preprint arXiv:2508.10875*, 2025.
- [36] Y. Du and I. Mordatch, “Implicit generation and modeling with energy based models,” *Advances in neural information processing systems*, vol. 32, 2019.
- [37] Y. Nesterov *et al.*, *Lectures on convex optimization*. Springer, 2018, vol. 137.
- [38] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” in *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, 2019, pp. 4171–4186.
- [39] S. Li, Y. Du, G. Van de Ven, and I. Mordatch, “Energy-based models for continual learning,” in *Conference on lifelong learning agents*. PMLR, 2022, pp. 1–22.

- [40] K. Sohn, D. Berthelot, N. Carlini, Z. Zhang, H. Zhang, C. A. Raffel, E. D. Cubuk, A. Kurakin, and C.-L. Li, “Fixmatch: Simplifying semi-supervised learning with consistency and confidence,” *Advances in neural information processing systems*, vol. 33, pp. 596–608, 2020.
- [41] A. A. Jafari, C. Ozcinar, and G. Anbarjafari, “Dynamic nested hierarchies: Pioneering self-evolution in machine learning architectures for lifelong intelligence,” *arXiv preprint arXiv:2511.14823*, 2025.
- [42] G. Delétang, A. Ruoss, P.-A. Duquenne, E. Catt, T. Genewein, C. Mattern, J. Grau-Moya, L. K. Wenliang, M. Aitchison, L. Orseau *et al.*, “Language modeling is compression,” *arXiv preprint arXiv:2309.10668*, 2023.
- [43] I. Loshchilov, F. Hutter *et al.*, “Fixing weight decay regularization in adam,” *arXiv preprint arXiv:1711.05101*, vol. 5, no. 5, p. 5, 2017.
- [44] M. Hardt and Y. Sun, “Test-time training on nearest neighbors for large language models,” *arXiv preprint arXiv:2305.18466*, 2023.
- [45] K. A. Wang, J. Shi, and E. B. Fox, “Test-time regression: a unifying framework for designing sequence models with associative memory,” *arXiv preprint arXiv:2501.12352*, 2025.
- [46] D. Guo, D. Yang, H. Zhang, J. Song, R. Zhang, R. Xu, Q. Zhu, S. Ma, P. Wang, X. Bi *et al.*, “Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning,” *arXiv preprint arXiv:2501.12948*, 2025.
- [47] X. Tian, S. Zhao, H. Wang, S. Chen, Y. Ji, Y. Peng, H. Zhao, and X. Li, “Think twice: Enhancing llm reasoning by scaling multi-round test-time thinking,” *arXiv preprint arXiv:2503.19855*, 2025.
- [48] S. McCallum, K. Arora, and J. Foster, “Reversible deep equilibrium models,” *arXiv preprint arXiv:2509.12917*, 2025.
- [49] M. Gabor, T. Piotrowski, and R. L. Cavalcante, “Positive concave deep equilibrium models,” *arXiv preprint arXiv:2402.04029*, 2024.
- [50] S. Gong, S. Agarwal, Y. Zhang, J. Ye, L. Zheng, M. Li, C. An, P. Zhao, W. Bi, J. Han *et al.*, “Scaling diffusion language models via adaptation from autoregressive models,” *arXiv preprint arXiv:2410.17891*, 2024.
- [51] J. Ou, S. Nie, K. Xue, F. Zhu, J. Sun, Z. Li, and C. Li, “Your absorbing discrete diffusion secretly models the conditional distributions of clean data,” *arXiv preprint arXiv:2406.03736*, 2024.
- [52] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou *et al.*, “Chain-of-thought prompting elicits reasoning in large language models,” *Advances in neural information processing systems*, vol. 35, pp. 24 824–24 837, 2022.
- [53] U. Neisser, “Cognitive psychology appleton-century-crofts new york,” *For personal use only—not for reproduction*, 1967.
- [54] A. Yuille and D. Kersten, “Vision as bayesian inference: analysis by synthesis?” *Trends in cognitive sciences*, vol. 10, no. 7, pp. 301–308, 2006.
- [55] B. J. Baars, “In the theatre of consciousness. global workspace theory, a rigorous scientific theory of consciousness,” *Journal of consciousness Studies*, vol. 4, no. 4, pp. 292–309, 1997.
- [56] S. Dehaene, L. Charles, J.-R. King, and S. Marti, “Toward a computational theory of conscious processing,” *Current opinion in neurobiology*, vol. 25, pp. 76–84, 2014.
- [57] L. G. McCoy, R. Swamy, N. Sagar, M. Wang, S. Bacchi, J. M. N. Fong, N. C. Tan, K. Tan, T. A. Buckley, P. Brodeur *et al.*, “Assessment of large language models in clinical reasoning: A novel benchmarking study,” *NEJM AI*, vol. 2, no. 10, p. A1dbp2500120, 2025.
- [58] C. V. Snell, J. Lee, K. Xu, and A. Kumar, “Scaling llm test-time compute optimally can be more effective than scaling parameters for reasoning,” in *The Thirteenth International Conference on Learning Representations*, 2025.
- [59] Y. Song, P. Dhariwal, M. Chen, and I. Sutskever, “Consistency models,” 2023.
- [60] Y. Liu, J. Yu, Y. Xu, Z. Li, and Q. Zhu, “A survey on transformer context extension: Approaches and evaluation,” *arXiv preprint arXiv:2503.13299*, 2025.