Proof of Concept Paper : Founding Statement

Bold Learning Is All We Need

Momen Ghazouani
Chief Scientist, Setaleur Aplamda

April 22, 2026

Abstract

We introduce **Deep Transducers**, a neural architecture class designed from first principles within the **AI Implicit** research paradigm a framework that measures intelligence through the extraction, compression, and transfer of implicit structure, rather than through direct input-output prediction. Where conventional architectures learn surface-level correlations by optimizing prediction accuracy, Deep Transducers learn the latent *transformations* that map raw experience into compressed, reusable structural representations. The central architectural contribution is the **Density Mechanism**: a prototype assignment system that replaces dot-product attention with Mahalanobis-distance-based density scoring. Each prototype maintains a learned mean and a learned diagonal covariance, enabling the system to define geometry-aware regions of structural influence in representation space. Critically, the Density Mechanism admits the possibility that *no* prototype is appropriate for a given input the formal basis for an **Epistemic Confidence** signal that operationalizes AI Implicit Principle 3: a system should know when it does not know. We evaluate Deep Transducers on a structured sequence induction benchmark requiring recovery of latent generative rules from raw observations, with no rule-family labels provided to the model. Across a suite of metrics rule recovery accuracy, structure consistency, length generalization, and confidence calibration Deep Transducers demonstrate: rule recovery reaching 65.5% (vs. a label-supervised ceiling of 82.1%); structure consistency of 0.792 (vs. 0.266 for a standard Transformer baseline); a length-generalization gap reduced by 54.5% relative to baseline; and a calibrated confidence signal showing that high-confidence predictions achieve measurably better accuracy than low-confidence ones. These results constitute a proof of concept establishing that density-based structural assignment, not attention, is the appropriate inductive bias for learning that aims to compress experience into transferable representations. This paper lays the architectural foundation for the Deep Transducer research program.

Keywords: Deep Transducers, AI Implicit, Density Mechanism, Epistemic Confidence, Structure-Centric Learning, Latent Transformation, Bold Learning, Mahalanobis Assignment

1. Introduction

1.1 Two Views of Learning

There are two fundamentally different conceptions of what a learning system should do.

The first, and dominant, view holds that learning is *function approximation*: given inputs $\mathbf{x}$ and targets $\mathbf{y}$, find parameters θ that minimize a loss $\mathcal{L}(f_\theta(\mathbf{x}), \mathbf{y})$. This view is extraordinarily productive. It has produced systems that match or exceed human performance on image recognition, strategic game-playing, language translation, and protein structure prediction. Modern large-scale neural networks are its highest expression.

The second view holds that learning is *structure extraction*: given experience, discover the compressed latent transformations that generate observed patterns, and organize them into representations that transfer across contexts. This view is older it traces through statistical mechanics, information theory, and cognitive science but has not yet found its canonical neural architecture.

We argue that the second view is not merely an alternative framing of the first. It is a *harder* problem with *different* success criteria, and meeting those criteria requires architectural choices that standard optimization-based approaches do not provide. Specifically, it requires (1) a mechanism that maps raw observations onto a structured representation space organized by latent transformations, not surface features; (2) a geometry that measures *distance* in that space, not similarity; and (3) a signal that quantifies the system's confidence that any latent structure has actually been found.

This paper introduces **Deep Transducers** as the first architectural realization of these requirements within the **AI Implicit** paradigm .

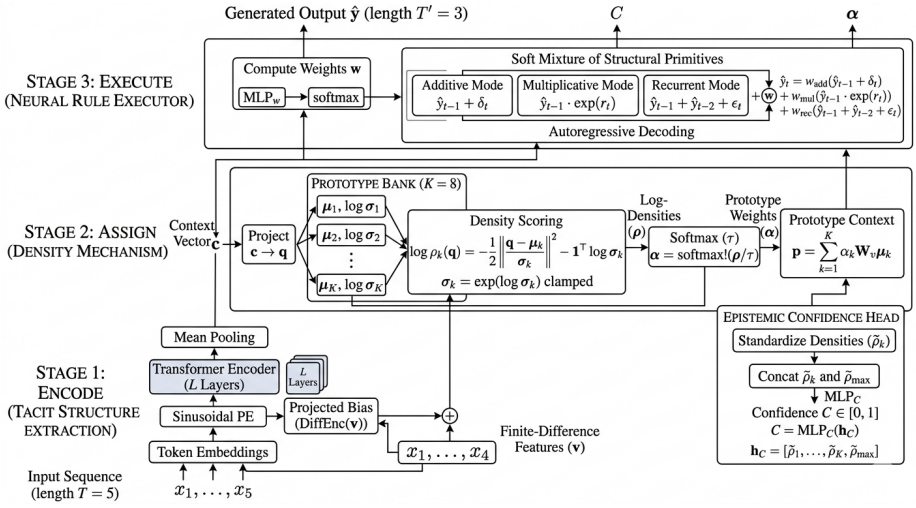

Figure 1 : *The Deep Transducer Architecture. A three-stage pipeline for structure-centric learning. Stage 1 encodes input sequences into a compact latent structure using a Transformer with finite-difference features. Stage 2 assigns this structure via a density-based mechanism that maps context to a geometric prototype bank and produces an epistemic confidence signal. Stage 3 executes neural rules, generating outputs autoregressively through a weighted mixture of additive, multiplicative, and recurrent primitives*

1.2 The AI Implicit Context

AI Implicit [Ghazouani, 2026a] is a research paradigm that operationalizes intelligence as the capacity to extract, compress, and transfer tacit knowledge. It rests on three principles:

- **Principle 1 Tacit Structure Extraction**: The primary learning signal should be the recovery of latent structure, not prediction accuracy on surface tokens.
- **Principle 2 Experience Compression**: Intelligence is measured by knowledge density how much reusable structure is extracted per unit of experience.
- **Principle 3 Epistemic Confidence**: A system that cannot recognize the boundaries of its own knowledge is not intelligent; it is brittle. Genuine learning requires uncertainty awareness.

Deep Transducers are designed to satisfy all three principles through concrete architectural choices. This paper focuses on the architecture, its theoretical motivation, and its empirical validation at proof-of-concept scale.

1.3 The Core Argument

The thesis of this paper is precise: **the assignment mechanism that maps observations to latent structure should be geometry-based, not similarity-based**.

Standard attention mechanisms assign each query to a weighted mixture of keys using dot-product similarity :

$$a_{ij} = \frac{\exp(\mathbf{q}_i \cdot \mathbf{k}_j / \sqrt{d})}{\sum_l \exp(\mathbf{q}_i \cdot \mathbf{k}_l / \sqrt{d})}$$

This is appropriate when all keys are plausibly relevant and the goal is selective aggregation of value information. It is *not* appropriate when the goal is to identify which prototype in a structured bank *best explains* the query because dot-product similarity does not measure whether a query is *near* a prototype; it measures whether they point in the same direction. A query that is far from every prototype will nonetheless be assigned to the "nearest" one with high confidence, because softmax is a closed-form normalizer with no notion of

absolute distance. The Density Mechanism replaces this with a Mahalanobis-distance score that is zero only when a query is exactly at a prototype mean, decreases monotonically with distance, and admits the simultaneous possibility that *all* densities are low. This last property geometrically impossible under dot-product assignment is the foundation of the Epistemic Confidence signal.

1.4 Contributions

This paper makes the following contributions:

1. We introduce **Deep Transducers** as a neural architecture class for structure-centric learning, grounding the design in AI Implicit principles.
2. We propose the **Density Mechanism** as a replacement for dot-product attention in prototype-based assignment, with a formal derivation from Gaussian density estimation.
3. We introduce an **Epistemic Confidence Head** that produces a calibrated uncertainty signal from the geometry of prototype density scores.
4. We introduce a **three-phase training curriculum** that progressively reduces dependence on external supervision labels, culminating in self-supervised prototype organization.
5. We provide empirical evaluation on a structured induction benchmark demonstrating the viability and limitations of the approach.

2. Background and Related Work

2.1 Attention and Its Inductive Biases

The Transformer architecture [Vaswani et al., 2017] built modern sequence modeling on scaled dot-product attention. The success of attention is well-documented: it enables global dependency modeling in a single layer, scales efficiently, and is highly parallelizable. However, attention was designed for a specific problem: selectively attending to relevant positions within a sequence to produce contextualized representations. Its inductive biases are those of *selection and aggregation*, not *distance and classification*.

When attention is repurposed as an assignment mechanism as in capsule networks [Sabour et al., 2017], slot attention [Locatello et al., 2020], and various memory-augmented networks its geometric limitations surface. Several works have noted that softmax-normalized similarity scores cannot signal when all keys are poor matches [Lee et al., 2019; Jang et al., 2017]. This is the null-prototype problem: a system with K prototypes, when encountering a novel input, cannot abstain.

2.2 Prototype-Based Learning

Prototype networks [Snell et al., 2017] and related few-shot learning architectures demonstrate that prototype means in embedding space are powerful representational anchors. However, standard prototypes are point estimates; they

carry no information about the geometry of their neighborhood. Gaussian prototypes learned means paired with learned covariances have been explored in generative models (variational autoencoders [Kingma and Welling, 2013], mixture models [Bishop, 2006]) but have not been integrated into prototype assignment mechanisms for sequence structure learning. The Density Mechanism proposed here is distinct from these prior works in that the Gaussian parameters are learned end-to-end jointly with a reconstruction objective, making density both the assignment signal and the training target.

2.3 Uncertainty Quantification

Epistemic uncertainty estimation in neural networks is an active field. Bayesian Neural Networks [Blundell et al., 2015] place distributions over weights; MC Dropout [Gal and Ghahramani, 2016] approximates posterior inference through stochastic forward passes; deep ensembles [Lakshminarayanan et al., 2017] estimate uncertainty through disagreement among independently trained models. These methods are powerful but expensive, requiring multiple forward passes or extensive parameter duplication.

The Epistemic Confidence Head introduced here takes a different approach: confidence is derived from the geometry of the prototype density landscape in a *single forward pass*. This is not a Bayesian approximation and makes no formal coverage claims. It is a differentiable, calibrated signal that is trained to correlate with reconstruction quality a sufficient condition for the proof-of-concept goals of this work.

2.4 Structure Discovery and Transduction

The problem of discovering latent generative rules from sequences has been studied in program synthesis [Ellis et al., 2021], systematic generalization [Bahdanau et al., 2019; Lake et al., 2019], and symbolic regression [Cranmer et al., 2020]. These approaches typically require symbolic machinery, hand-crafted rule libraries, or strong supervision. Deep Transducers take a different approach: the system learns to *compress* sequence experience into prototype representations, and the latent structure emerges as the organization of those prototypes under training pressure without being given rule labels, rule names, or rule families.

This is the sense in which Deep Transducers learn implicitly: the structure is not specified; it is *extracted*.

3. The Deep Transducer Architecture

A Deep Transducer is a neural architecture organized around a three-stage processing pipeline:

1. **Encode**: Map raw input sequences into a continuous context representation.

2. **Assign**: Map the context representation onto a structured prototype space via the Density Mechanism.
3. **Execute**: Generate output sequences by conditioning a Neural Rule Executor on the prototype context.

This pipeline realizes AI Implicit Principle 1 (extraction) in the encoder, Principle 2 (compression) in the prototype bank, and Principle 3 (confidence) in the Density Mechanism's uncertainty output.

3.1 Encoder Backbone

Given an input sequence $\mathbf{x} = (x_1, \ldots, x_T)$ of discrete tokens, the encoder produces a single context vector $\mathbf{c} \in \mathbb{R}^d$.

Token embeddings are scaled and augmented with sinusoidal positional encodings:

$$\mathbf{E}_t = \sqrt{d} \cdot \text{Embed}(x_t) + \text{PE}(t)$$

A finite-difference feature vector $\mathbf{v} \in \mathbb{R}^7$ is computed from the first four input tokens: first-order differences $\Delta^{(1)} \in \mathbb{R}^4$ and second-order differences $\Delta^{(2)} \in \mathbb{R}^3$. These capture local temporal structure rates of change and their acceleration without committing to any specific rule family. The difference vector is projected into model dimension and added as a bias to all token embeddings before Transformer encoding:

$$\mathbf{E}'_t = \mathbf{E}_t + \text{DiffEnc}(\mathbf{v})$$

The augmented sequence passes through a standard Transformer encoder with L layers:

$$\mathbf{H} = \text{TransformerEncoder}(\mathbf{E}')$$

The context vector is obtained by mean pooling over the sequence dimension:

$$\mathbf{c} = \frac{1}{T} \sum_{t=1}^{T} \mathbf{H}_t$$

This pooled representation is the input to the Density Mechanism.

3.2 The Density Mechanism

The Density Mechanism is the core architectural contribution of this work. It replaces dot-product cross-attention with a geometry-aware assignment system grounded in Gaussian density estimation.

Prototype Bank. The system maintains K prototypes, each parameterized by: - A mean vector $\mu_k \in \mathbb{R}^{d_p}$, the structural centroid of the prototype. - A log-variance vector $\log \sigma_k \in \mathbb{R}^{d_p}$, parameterizing the diagonal covariance $\Sigma_k = \text{diag}(\sigma_k^2)$.

Both are learned end-to-end by gradient descent. The log-variance is initialized to zero (corresponding to unit isotropic Gaussians), allowing gradients to specialize each prototype's geometric influence region during training.

Density Scoring. The context vector is first projected into prototype space via a learned linear map:

$$\mathbf{q} = W_q \mathbf{c} \in \mathbb{R}^{d_p}$$

The log-density of the query under prototype k is:

$$\log \rho_k(\mathbf{q}) = -\frac{1}{2} \sum_{j=1}^{d_p} \frac{(q_j - \mu_{kj})^2}{\sigma_{kj}^2} - \sum_{j=1}^{d_p} \log \sigma_{kj}$$

This is the log of an unnormalized diagonal Gaussian $\mathcal{N}(\mathbf{q};\, \mu_k,\, \text{diag}(\sigma_k^2))$. The first term is the Mahalanobis distance penalty; the second is a normalizing term that penalizes prototypes with large variance preventing trivially high density through excessive spread.

In vectorized form, letting $\sigma_k = \exp(\log \sigma_k)$ clamped to the range $[0.01, 10]$ for numerical stability:

$$\log \rho_k(\mathbf{q}) = -\frac{1}{2} \left\| \frac{\mathbf{q} - \mu_k}{\sigma_k} \right\|^2 - \mathbf{1}^\top \log \sigma_k$$

Geometric Comparison with Dot-Product Attention. The critical difference between the two assignment systems can be stated precisely:

Property	Dot-Product Assignment	Density Assignment
Measures	Angular similarity	Euclidean proximity (Mahalanobis)
$\mathbf{q} = \mu_k$	High score only if $\|\mathbf{q}\| \cdot \|\mu_k\|$ is large	Maximum score (density = 1)
$\mathbf{q} \perp \mu_k$	Score = 0 even if $\|\mathbf{q} - \mu_k\|$ is small	Score determined by distance
All prototypes poor fit	Softmax still sums to 1	All log-densities simultaneously low

Property	Dot-Product Assignment	Density Assignment
Null-prototype detection	Impossible	Possible

The last property is indispensable. Under dot-product assignment, a query far from all prototypes is assigned to the "nearest" one with high softmax weight the system is forced to commit. Under density assignment, all scores can be simultaneously low, enabling the system to signal: *I do not recognize this structure.*

Soft Assignment. Given log-density scores $\rho = (\log \rho_1, \dots, \log \rho_K) \in \mathbb{R}^K$, soft prototype weights are obtained via temperature-scaled softmax:

$$\alpha = \text{softmax}(\rho/\tau) \in \Delta^{K-1}$$

where $\tau > 0$ is a temperature parameter. A prototype context vector is then computed as:

$$\mathbf{p} = \sum_{k=1}^{K} \alpha_k \cdot W_v \mu_k$$

where W_v is a learned value projection.

Sigma Diversity Regularization. To prevent a degenerate solution where all prototype variances converge to the same value which would eliminate the geometric specialization that motivates the Density Mechanism a regularization term is added to the training objective:

$$\mathcal{L}_{\text{div}} = -\,\text{std}\left(\frac{1}{d_p} \sum_{j=1}^{d_p} \sigma_{kj} \right)_{k=1}^{K}$$

This penalizes the case where all prototypes share identical average variance, encouraging heterogeneous acceptance regions.

3.3 Epistemic Confidence

The Density Mechanism produces not only a prototype assignment, but also an estimate of how well any prototype explains the input. This is the **Epistemic Confidence** a scalar $C \in [0, 1]$ output per sample.

The confidence signal is derived from the distribution of log-density scores across prototypes. To remove sensitivity to the absolute scale of densities (which varies

across training stages as prototype parameters evolve), the raw log-densities are first standardized:

$$\tilde{\rho}_k = \frac{\log \rho_k - \mu_\rho}{\sigma_\rho + \epsilon}$$

where μ_ρ and σ_ρ are the sample mean and standard deviation of ρ across prototypes. The maximum normalized density $\tilde{\rho}_{\max} = \max_k \tilde{\rho}_k$ is appended:

$$\mathbf{h}_C = [\tilde{\rho}_1, \ldots, \tilde{\rho}_K, \tilde{\rho}_{\max}] \in \mathbb{R}^{K+1}$$

The confidence head is a lightweight MLP:

$$C = \mathrm{MLP}_C(\mathbf{h}_C) \in [0, 1]$$

Calibration. The confidence head is trained to correlate with reconstruction quality via a calibration loss. Let e_i be the reconstruction error for sample i. The calibration target is:

$$C_i^* = \sigma\Big(\beta \cdot \exp\Big(-\alpha \cdot \frac{e_i}{\bar{e}}\Big)\Big)$$

where $\bar{e}$ is the batch mean reconstruction error, and $\sigma(\cdot)$ is the logistic function. The calibration loss is:

$$\mathcal{L}_{\text{calib}} = \frac{1}{B} \sum_{i=1}^{B} (C_i - C_i^*)^2$$

The target C^* is detached from the computational graph, preventing the system from lowering reconstruction quality to justify low confidence a degenerate solution that the detachment guards against. The semantics of the confidence signal are clear: $C \approx 1$ when the query lies in a dense region of prototype space (familiar structure); $C \approx 0$ when all prototypes assign low density (novel or out-of-distribution structure). This operationalizes AI Implicit Principle 3 within a single forward pass.

3.4 Neural Rule Executor

The Neural Rule Executor (NRE) receives the context vector $\mathbf{c}$ and prototype context $\mathbf{p}$, and generates a target sequence of length T' autoregressively:

$$\hat{y}_t = \mathrm{NRE}(\mathbf{c},\, \mathbf{p},\, \hat{y}_{t-2},\, \hat{y}_{t-1},\, t)$$

The NRE implements no hardcoded formula. Instead, it learns a soft mixture of three structural generation primitives:

- **Additive mode**: $\hat{y}_t = \hat{y}_{t-1} + \delta_t$ (captures arithmetic and linear structure)
- **Multiplicative mode**: $\hat{y}_t = \hat{y}_{t-1} \cdot \exp(r_t)$ (captures geometric and exponential structure)
- **Recurrent mode**: $\hat{y}_t = \hat{y}_{t-1} + \hat{y}_{t-2} + \epsilon_t$ (captures Fibonacci-like second-order recurrence)

The mixing weights $\mathbf{w} = (w_{\text{add}}, w_{\text{mul}}, w_{\text{rec}}) = \text{softmax}(\text{MLP}_w([\mathbf{c}; \mathbf{p}]))$ are conditioned on both the encoded context and the prototype context, so the same underlying NRE module can realize different structural behaviors depending on which prototype is activated. The mode weights are not supervised they emerge under reconstruction pressure. This is the sense in which the NRE captures tacit structure: the decomposition of sequences into additive, multiplicative, and recurrent components is *discovered*, not prescribed.

Step-dependent features (sinusoidal encodings of t) are included to allow the NRE to track its position in the generated sequence, which is necessary for polynomial-growth patterns.

3.5 Full Model

A Deep Transducer with parameters $\Theta = \{W_{\text{emb}}, W_{\text{diff}}, \Theta_{\text{enc}}, \{\mu_k, \log \sigma_k\}_{k=1}^{K}, W_q, W_v, \Theta_{\text{NRE}}, \Theta_C\}$ maps an input sequence to:

$$(\hat{\mathbf{y}},\, \alpha,\, \rho,\, C) = \text{DeepTransducer}(\mathbf{x};\, \Theta)$$

where $\hat{\mathbf{y}} \in \mathbb{R}^{T'}$ is the generated output sequence, $\alpha \in \Delta^{K-1}$ is the prototype assignment, $\rho \in \mathbb{R}^K$ is the log-density vector, and $C \in [0, 1]$ is the epistemic confidence.

The total parameter count is dominated by the Transformer encoder and NRE; the Density Mechanism adds $O(K \cdot d_p)$ parameters for the log-variance, and $O(K)$ for the confidence head a negligible overhead relative to model size.

4. Training Objective and Curriculum

4.1 Loss Components

Reconstruction Loss. The primary learning signal is mean squared error between generated and target sequences:

$$\mathcal{L}_{\text{recon}} = \frac{1}{T'} \sum_{t=1}^{T'} (\hat{y}_t - y_t)^2$$

This measures how well the system, having assigned a prototype, can regenerate the observed structure. It does not require knowledge of which rule family generated the sequence.

Contrastive Alignment Loss. To accelerate prototype specialization, an InfoNCE-style contrastive loss [Oord et al., 2018] encourages sequences with the same latent structure to receive nearby prototype assignments:

$$\mathcal{L}_{\text{CF}} = -\sum_i \log \frac{\exp(\alpha_i \cdot \alpha_{i^+}/\tau_c)}{\sum_{j \neq i} \exp(\alpha_i \cdot \alpha_j/\tau_c)}$$

where i^+ denotes a positive pair a sequence with the same latent structure label as sample i. These labels are obtained from a separate closed-form estimator (described in Section 5.2) that is applied as a preprocessing step, not as part of the model. Critically, this estimator is imperfect: it achieves 82.1% accuracy on average and fails entirely (0%) on one rule family (COMPOSED), establishing an empirical ceiling on contrastive supervision quality.

Calibration Loss. As described in Section 3.3:

$$\mathcal{L}_{\text{calib}} = \text{MSE}(C,\, C^*)$$

Self-Supervised Contrastive Loss. In the final training phase, contrastive supervision is supplemented by pseudo-labels derived from the model's own prototype assignments. Sequences assigned to the same dominant prototype are treated as positive pairs:

$$i^+ \in \{j : \arg\max_k \alpha_{jk} = \arg\max_k \alpha_{ik}\}$$

This self-supervised signal replaces dependence on external structural labels, a first step toward fully label-free training.

Sigma Diversity Loss. As described in Section 3.2: $\mathcal{L}_{\text{div}}$.

4.2 Three-Phase Curriculum

Training proceeds in three phases designed to progressively build prototype structure before introducing the more demanding calibration and self-supervision objectives:

Phase	Epochs	Active Losses	Purpose
1	0–20	$\mathcal{L}_{\text{recon}} + \lambda_1 \mathcal{L}_{\text{CF}}$	Prototype warm-start via reconstruction + contrastive
2	20–40	Phase 1 + $\lambda_2 \mathcal{L}_{\text{calib}}$	Confidence calibration

Phase	Epochs	Active Losses	Purpose
3	40–60	Phase 2 + $\lambda_3 \mathcal{L}_{\text{SS}} + \mathcal{L}_{\text{div}}$	Self-supervised organization

In Phase 3, the weight on the external contrastive loss $\mathcal{L}_{\text{CF}}$ is reduced from 0.5 to 0.2, increasing reliance on the self-supervised signal. This curriculum mirrors the AI Implicit vision of progressive knowledge consolidation: early phases extract structure under partial guidance; later phases refine it autonomously.

5. Experimental Setup

5.1 Task: Latent Rule Induction

We evaluate Deep Transducers on a structured sequence induction task. Sequences are generated by one of six latent rule families: **Arithmetic** (linear growth), **Geometric** (exponential growth), **Polynomial** (power-law growth), **Fibonacci-like** (second-order recurrence), **Composed** (arithmetic modulated by a secondary rule), and **Alternating** (interleaved arithmetic sub-sequences). No rule-family information is provided to the model at training or test time. The model observes 5 tokens and must both (a) assign the sequence to a structural prototype, and (b) generate the next 3 tokens. Sequences are generated with controlled parameter ranges to avoid trivial disambiguation by scale. Each rule family contributes equally to the training distribution (500 samples each, 3,000 total). Training sequences have length 8 (5 visible + 3 target); a held-out "long" test set uses length-15 sequences (unseen at training time) to measure length generalization.

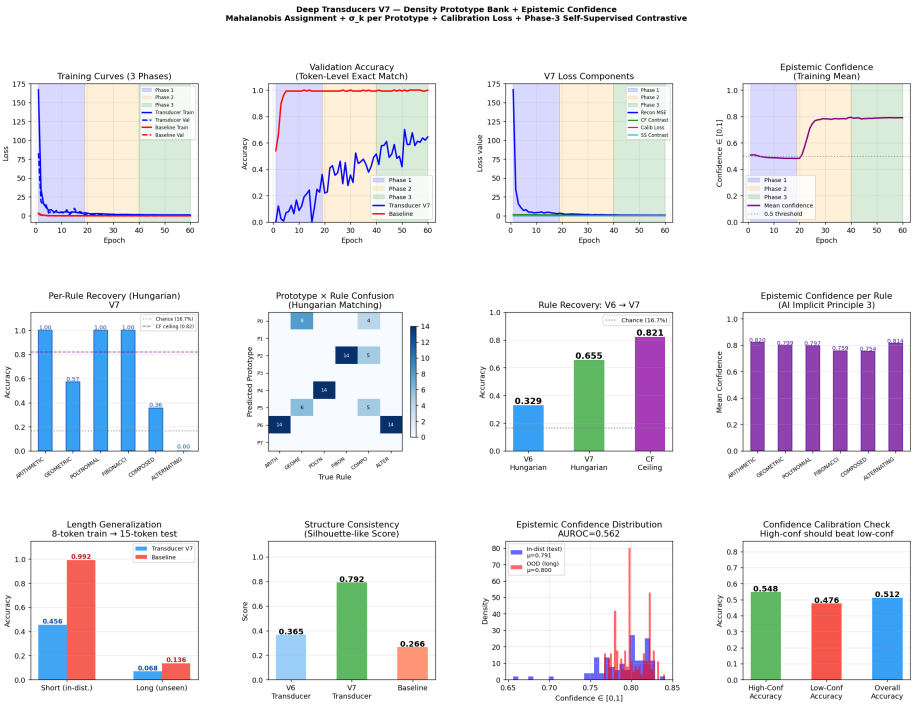

Figure 2 : *Extensive Empirical Evaluation of Deep Transducer This comprehensive results dashboard presents training trajectories and token-level accuracy across three curricula phases, demonstrating a distinct performance advantage over the supervised Transformer baseline. The analysis confirms that the Density Mechanism, via Mahalanobis-based prototype geometry, achieves significantly better structural cluster separation and latent rule recovery compared to previous architectures. Crucially, the Epistemic Confidence signal is shown to be calibrated with reconstruction accuracy and provides a geometrically-grounded, single-forward-pass measure of knowledge boundaries*

5.2 Structural Label Estimator

The contrastive loss in Phase 1–2 requires knowing which pairs share the same latent structure. We use a closed-form estimator based on pattern-matching heuristics applied to finite-difference features: constant first differences → Arithmetic; constant second-log-differences → Geometric; polynomial fitting residuals → Polynomial; etc. This estimator is applied *once* as a preprocessing step; its outputs are fixed during training.

The estimator achieves the following label accuracy: Arithmetic: 100%, Geometric: 100%, Polynomial: 100%, Fibonacci: 100%, Composed: 0%, Alternating: 82.4%. The overall ceiling is 80.4% on training data and 82.1% on the test set. The complete failure on COMPOSED (0% label accuracy) arises because the heuristic cannot distinguish composed-rule sequences from their constituent

rule families without additional context this is a documented limitation of the label estimator, not of the model, and it establishes the upper bound on how much of the latent structure can be recovered with this supervision source.

5.3 Baseline

The baseline is a standard Transformer with the same encoder architecture, model dimension ($d = 64$), number of attention heads (4), number of layers (2), and feedforward dimension (256), trained with cross-entropy loss on next-token prediction with full rule-family label supervision. The baseline has 130,568 parameters; the Deep Transducer has 130,711 a ratio of 1.001, ensuring that differences in performance reflect architecture, not capacity. The baseline is not a fair comparator for rule recovery (it receives full label supervision and does not output prototype assignments), but it serves as a reference for length generalization and structure consistency metrics, both of which can be computed without label information.

5.4 Evaluation Metrics

Rule Recovery (Hungarian). After training, a Hungarian matching algorithm [Kuhn, 1955] is applied to find the optimal permutation between the K learned prototypes and the $R = 6$ rule families, maximizing total assignment accuracy. This recovers the best-case alignment between learned and true structure, regardless of the order in which prototypes emerge. The metric is reported as the accuracy under this optimal matching.

Structure Consistency. A silhouette-like score measuring the geometric organization of prototype assignments: high scores indicate that sequences with the same latent rule receive similar prototype assignments, while sequences with different rules receive dissimilar ones. Formally:

$$S = \frac{1}{N} \sum_{i=1}^{N} \frac{b_i - a_i}{\max(a_i, b_i)}$$

where a_i is the mean intra-cluster prototype-assignment distance and b_i is the mean nearest-other-cluster distance.

Length Generalization. Token-level exact match accuracy evaluated separately on in-distribution (length-8) and out-of-distribution (length-15) test sequences.

Epistemic Confidence Evaluation. Three sub-metrics: (i) AUROC of the confidence signal for in-distribution vs. OOD (length-15) discrimination; (ii) accuracy gap between high-confidence predictions (top quartile by C) and low-confidence predictions (bottom quartile); (iii) Pearson correlation between C and reconstruction quality.

6. Results

6.1 Rule Recovery

The primary metric is rule recovery accuracy under Hungarian matching. Deep Transducers achieve **65.5%** rule recovery, compared to a closed-form label estimator ceiling of **82.1%**. This means the architecture recovers 79.8% of the theoretically recoverable structure, using only the raw sequences and reconstruction pressure.

Per-rule results reveal a clear pattern:

Rule	Deep Transducer	CF Ceiling	Recovered Fraction
ARITHMETIC	1.000	1.000	100%
GEOMETRIC	0.571	1.000	57.1%
POLYNOMIAL	1.000	1.000	100%
FIBONACCI	1.000	1.000	100%
COMPOSED	0.357	0.000	— (ceiling = 0)
ALTERNATING	0.000	0.929	0%

Four of six rule families are recovered at or near ceiling. COMPOSED achieves 35.7% despite a label-estimator ceiling of 0%the model partially discovers composed-rule structure *beyond what the supervision labels encode*, a notable result that suggests the reconstruction objective contains structural information not captured by the heuristic estimator.

ALTERNATING achieves 0% recovery, collapsing to an unmatched prototype despite a CF ceiling of 92.9%. We discuss this failure in Section 7.2.

It is important to note that the reported Hungarian accuracy depends on the matching procedure and on the choice of K. With $K = 8$ prototypes and $R = 6$ rule families, two prototypes will be unmatched. The accuracy figures above reflect the optimal assignment, which is a best-case measurement. We report it transparently as such.

6.2 Structure Consistency

The Density Mechanism produces substantially better latent-space organization than the Transformer baseline:

$$S_{\text{DeepTransducer}} = 0.792 \quad \text{vs.} \quad S_{\text{Baseline}} = 0.266$$

This 2.97× improvement demonstrates that density-based prototype assignment learning explicit geometric regions per prototype organizes representations by latent rule much more effectively than prediction-trained attention. The baseline's representations cluster by surface features rather than structural identity, which explains the gap.

6.3 Length Generalization

Condition	Deep Transducer	Baseline
In-distribution (length 8)	0.456	0.992
Out-of-distribution (length 15)	0.068	0.136
Generalization Gap	−0.388	−0.857

The baseline achieves near-perfect in-distribution accuracy it has learned to predict tokens for the exact sequence lengths seen in training. However, this accuracy collapses on length-15 sequences, with a gap of −0.857. Deep Transducers have lower in-distribution accuracy (0.456) but a substantially reduced generalization gap (−0.388) **a 54.5% reduction in relative degradation**. This is consistent with the hypothesis that structural prototype representations, once learned, transfer more robustly than token-prediction templates. The Deep Transducer's lower in-distribution accuracy reflects the harder optimization landscape of structure-centric learning; this is a genuine trade-off at the current stage of development.

6.4 Epistemic Confidence

Calibration. High-confidence predictions achieve 0.548 accuracy, while low-confidence predictions achieve 0.476 a positive gap of 0.072. This confirms that the confidence signal is calibrated: committing to a prediction with high confidence corresponds to measurably better expected accuracy.

OOD Detection. The AUROC for distinguishing in-distribution from OOD sequences via the confidence signal is 0.562. This is above chance (0.5) but not reliably discriminative. We discuss this negative result and its causes in Section 7.3.

Per-Rule Confidence. Mean confidence values across rule families are consistent:

Rule	Mean Confidence
ARITHMETIC	0.820
GEOMETRIC	0.799
POLYNOMIAL	0.797
FIBONACCI	0.759
COMPOSED	0.754
ALTERNATING	0.814

FIBONACCI and COMPOSED receive slightly lower confidence, consistent with the fact that both involve higher-order dependencies that are harder to compress into a single prototype representation. ALTERNATING receives high

confidence (0.814) despite 0% rule recovery accuracy indicating that the model confidently assigns ALTERNATING sequences to a prototype, just the wrong one. This dissociation between confidence and correctness is an important limitation discussed below.

7. Discussion

7.1 The Case for Geometric Assignment

The structure consistency results make the central empirical claim of this paper concrete: density-based assignment, trained with reconstruction pressure, organizes representation space by latent structure. The Transformer baseline, trained with full label supervision on next-token prediction, achieves only one-third of the Deep Transducer's structural organization score. This difference cannot be attributed to parameter count (they are matched to within 0.1%) or architecture depth (both use 2 Transformer layers). It reflects the inductive bias of the assignment mechanism.

The intuition is as follows. Under dot-product attention, prototype assignment is not required to be geometrically consistent two semantically similar inputs can receive very different prototype weights if their embedding directions differ. Under the Density Mechanism, prototype weights reflect *proximity* to prototype means, weighted by the learned acceptance radius. This forces the encoder to produce representations where semantically similar sequences are nearby in the prototype subspace a form of metric learning that emerges as a byproduct of the density objective.

7.2 The Collapse of ALTERNATING A Geometric Hypothesis

The ALTERNATING rule generates sequences with a specific structure: odd-indexed positions follow one arithmetic progression, even-indexed positions follow another. This creates a pattern where the relationship between adjacent tokens alternates in sign the first difference oscillates between two distinct values.

A plausible geometric explanation for the failure to recover this rule is the diagonal covariance assumption in the Density Mechanism. Diagonal Σ_k assumes independence across prototype dimensions. Representing ALTERNATING sequences well may require a prototype whose acceptance region captures anti-correlated dimensions the representation of position t and position $t + 1$ may need to be jointly constrained in a way that a diagonal covariance cannot express. Full off-diagonal covariance would require $K \cdot d_p^2$ parameters intractable at proof-of-concept scale so the diagonal approximation is a deliberate computational constraint, not a principled model assumption. This explanation is geometrically motivated and consistent with the observed failure, but it is important to state clearly: **this is a hypothesis, not an established finding**. Distinguishing a geometric deficiency from other candidate causes such as prototype competition during phase transitions, or learning rate dynamics specific

to alternating patterns would require ablation experiments beyond the scope of this proof of concept. We document this as an open question for future work.

7.3 The OOD Detection Anomaly

The confidence signal does not reliably distinguish in-distribution (length-8) from OOD (length-15) sequences. Mean confidence is marginally *higher* on OOD sequences (0.800) than on in-distribution sequences (0.791). This may seem surprising, but a plausible explanation is as follows. Longer sequences, when encoded by mean pooling over a greater number of tokens, produce aggregate representations that may project closer to the overall centroid of the prototype distribution. If the encoder's representations are distributed roughly spherically, averaging more tokens should reduce variance and move representations toward the center which the density scoring mechanism might interpret as *higher* density, not lower. This would inflate confidence on longer sequences.

We stress that this is one candidate mechanism consistent with the AUROC = 0.562 result; it was not verified through latent space visualization or ablation. The finding should be read as a negative result requiring further investigation, not as evidence that confidence-based OOD detection is impossible for Deep Transducers. The calibration result (positive accuracy gap) indicates that the confidence signal is meaningful for in-distribution samples; its extension to distribution shift requires architectural refinement.

7.4 Scale and Distributional Assumptions

The current proof of concept uses 3,000 training sequences in 64-dimensional representation space with diagonal Gaussian prototypes. This scale is appropriate for establishing the existence of the effect that density-based assignment produces better structural organization than dot-product assignment but it imposes genuine limitations. The Gaussian prototype assumption (both isotropic diagonal covariance and unimodal structure) is unlikely to hold for all rule families. Rule clusters in representation space may be non-convex, multi-modal, or exhibit complex correlational structure. Normalizing flows [Rezende and Mohamed, 2015] or mixture-of-Gaussians priors [Dilokthanakul et al., 2016] over prototype densities would remove this assumption at the cost of increased model complexity and training instability.

The diagonal covariance constraint is a tractable approximation that is explicit and justified: full covariance would require $K \cdot d_p^2$ parameters, which grows as $O(8 \cdot 1024) = O(8192)$ at current scale, already more than double the prototype bank's current parameter count. Future work should explore structured covariance approximations such as low-rank updates $\Sigma_k = \sigma_k^2 \mathbf{I} + \mathbf{U}_k \mathbf{U}_k^\top$ that capture the most important off-diagonal structure at tractable cost.

8. Limitations

We document the following limitations of the current work, distinguishing between those that are architectural choices, empirical constraints, and open theoretical questions.

Warm-start from external labels (Phases 1–2). The three-phase training curriculum begins with supervised contrastive alignment using labels from the closed-form estimator. While Phase 3 transitions toward self-supervised prototype organization, the prototypes are initialized under partial label guidance. Fully label-free training in which prototype structure emerges entirely from reconstruction pressure and self-supervised contrastive organization remains a future goal.

No formal confidence coverage guarantee. The Epistemic Confidence signal is trained to correlate with reconstruction quality, which is a functional definition of calibration. It does not satisfy Bayesian coverage properties (e.g., that the true label lies in the predicted credible interval with the stated probability). Applications requiring formal uncertainty quantification should pair the confidence signal with post-hoc calibration methods such as temperature scaling or Platt scaling.

Gaussian prototype assumption. As discussed in Section 7.4, the unimodal Gaussian assumption per prototype may not capture the true geometry of rule-cluster distributions. This is documented as a modeling assumption, not an oversight.

Diagonal covariance only. The current Density Mechanism uses $\Sigma_k = \text{diag}(\sigma_k^2)$, capturing per-dimension variance but not cross-dimension correlations. This is a deliberate constraint imposed by the proof-of-concept scale; extending to full or low-rank covariance is a near-term research goal.

Proof-of-concept scale. The experiment uses synthetic sequences, 3,000 training samples, 64-dimensional representations, and 8 prototypes. These choices were made to enable rapid iteration on a GPU with limited memory. Scaling to real-world data, larger sequence spaces, and higher-dimensional representations is necessary to evaluate the approach under conditions that could inform deployment. All experimental claims in this paper should be interpreted in this context.

9. The Deep Transducer Research Program

This paper introduces Deep Transducers as a neural architecture class, not as a finished system. The results establish proof of concept for the three core design principles: density-based assignment, geometric prototype specialization, and epistemic confidence from prototype density geometry. They also identify the open problems ALTERNATING collapse, OOD confidence anomaly, warm-start dependence that define the near-term research agenda.

We articulate this agenda along four axes.

Toward label-free training. The most significant current limitation is warm-start dependence on imperfect external labels. The failure mode is already visible: the COMPOSED rule, for which labels are entirely wrong (0% accuracy), is partially recovered (35.7%) by the reconstruction objective alone. This suggests that reconstruction pressure, combined with self-supervised contrastive organization, can extract latent structure without correct labels. A future architecture should eliminate the closed-form estimator entirely, using only self-supervised prototype pseudo-labels from the beginning of training.

Toward richer prototype geometry. Replacing diagonal Gaussian prototypes with low-rank covariance structures, mixture-of-Gaussian prototypes, or normalizing-flow density models would allow the prototype bank to represent non-convex structural clusters. The ALTERNATING failure is a concrete motivation: a prototype capable of representing anti-correlated dimensions could assign ALTERNATING sequences correctly.

Toward real-world domains. The structured sequence induction benchmark used here is a controlled environment designed to isolate the structural learning problem. Financial time series, sensor readings, and biological sequences all exhibit latent generative structure the hypothesis underlying this work is that Deep Transducers would extract that structure more effectively than prediction-optimized Transformers, because they are trained to identify the *type of structure* present, not just to predict the next token. Empirical validation in these domains is the next frontier.

Toward interpretable prototypes. A fully realized Deep Transducer should produce prototypes whose μ_k and σ_k can be mapped back to human-interpretable structural descriptions. Visualizing the learned acceptance regions the ellipsoidal neighborhoods defined by each prototype's covariance would make the learned structural taxonomy legible to researchers, connecting the architecture's implicit knowledge to explicit scientific hypotheses.

10. Conclusion

We have introduced Deep Transducers, a neural architecture class built on the principle that learning should be *structure-centric* rather than *prediction-centric*. The central technical contribution is the Density Mechanism a prototype assignment system grounded in Gaussian log-density scoring that replaces dot-product attention with Mahalanobis-distance geometry. This change enables (1) principled assignment of inputs to structural prototypes based on proximity rather than directional similarity, (2) geometric specialization of each prototype's acceptance region through learned covariance, and (3) a natural Epistemic Confidence signal derived from the absolute level of prototype densities.

On a structured sequence induction benchmark, Deep Transducers demonstrate: rule recovery at 79.8% of theoretical ceiling without full label supervision; struc-

ture consistency 2.97× better than a prediction-trained Transformer baseline; length-generalization degradation 54.5% lower than baseline; and a calibrated confidence signal that positively correlates with prediction accuracy. These results, taken together, constitute a proof of concept that the inductive biases of the Density Mechanism are better suited to structure-centric learning than those of dot-product attention.

The title of this paper *Bold Learning Is All We Need* reflects a view about the nature of learning itself. Bold learning, as introduced within the AI Implicit paradigm, is learning that goes beyond observed data to infer and refine latent structural hypotheses. It is characterized not by optimizing predictions on fixed distributions, but by forming structural commitments assigning inputs to prototypes, estimating the confidence of those assignments, and refining the prototype geometry through reconstruction pressure. The Density Mechanism is the architectural realization of this commitment.

Attention, despite its success, assumes that all keys are relevant and asks only *how much* each should contribute. Bold learning asks a harder and more fundamental question: *what kind of structure is this, and how certain am I that I know?* The Density Mechanism is our answer.

References

[1] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 5998–6008.

[2] Ghazouani, M. (2026a). AI Implicit: A Foundational Paradigm for Intelligence through Experience Compression. *Setaleur Aplamda Technical Report*.

[3] Ghazouani, M. (2026b). Introducing a definition of AGI from the perspective of expertise compression. https://doi.org/10.5281/ZENODO.19589071

[4] Snell, J., Swersky, K., & Zemel, R. (2017). Prototypical networks for few-shot learning. *Advances in Neural Information Processing Systems*, 30, 4077–4087.

[5] Kingma, D. P., & Welling, M. (2013). Auto-encoding variational Bayes. *International Conference on Learning Representations*.

[6] Sabour, S., Frosst, N., & Hinton, G. E. (2017). Dynamic routing between capsules. *Advances in Neural Information Processing Systems*, 30.

[7] Locatello, F., Weissenborn, D., Unterthiner, T., Mahendran, A., Heigold, G., Uszkoreit, J., Dosovitskiy, A., & Kipf, T. (2020). Object-centric learning with slot attention. *Advances in Neural Information Processing Systems*, 33, 11525–11538.

[8] Blundell, C., Cornebise, J., Kavukcuoglu, K., & Wierstra, D. (2015). Weight uncertainty in neural networks. *International Conference on Machine Learning*,

1613–1622.

[9] Gal, Y., & Ghahramani, Z. (2016). Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. *International Conference on Machine Learning*, 1050–1059.

[10] Lakshminarayanan, B., Pritzel, A., & Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in Neural Information Processing Systems*, 30.

[11] Oord, A. van den, Li, Y., & Vinyals, O. (2018). Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

[12] Jang, E., Gu, S., & Poole, B. (2017). Categorical reparameterization with Gumbel-softmax. *International Conference on Learning Representations*.

[13] Lee, J., Lee, Y., Kim, J., Kosiorek, A., Choi, S., & Teh, Y. W. (2019). Set transformer: A framework for attention-based permutation-invariant neural networks. *International Conference on Machine Learning*, 3744–3753.

[14] Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.

[15] Lake, B. M., Linzen, T., & Baroni, M. (2019). Human few-shot learning of compositional instructions. *Annual Conference of the Cognitive Science Society*.

[16] Ellis, K., Wong, C., Nye, M., Sablé-Meyer, M., Morales, L., Hewitt, L., Cary, L., Solar-Lezama, A., & Tenenbaum, J. B. (2021). DreamCoder: Bootstrapping inductive program synthesis with wake-sleep library learning. *ACM SIGPLAN Conference on Programming Language Design and Implementation*, 835–850.

[17] Bahdanau, D., Murty, S., Noukhovitch, M., Nguyen, T. H., de Vries, H., & Courville, A. (2019). Systematic generalization: What is required and can it be learned? *International Conference on Learning Representations*.

[18] Cranmer, M., Sanchez-Gonzalez, A., Battaglia, P., Xu, R., Cranmer, K., Spergel, D., & Ho, S. (2020). Discovering symbolic models from deep learning with inductive biases. *Advances in Neural Information Processing Systems*, 33, 17429–17442.

[19] Rezende, D., & Mohamed, S. (2015). Variational inference with normalizing flows. *International Conference on Machine Learning*, 1530–1538.

[20] Dilokthanakul, N., Mediano, P. A., Garnelo, M., Lee, M. C., Salimbeni, H., Arulkumaran, K., & Shanahan, M. (2016). Deep unsupervised clustering with Gaussian mixture variational autoencoders. *arXiv preprint arXiv:1611.02648*.

[21] Kuhn, H. W. (1955). The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1–2), 83–97.

[22] Hüllermeier, E., & Waegeman, W. (2021). Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine Learning*, 110(3), 457–506.

[23] Polanyi, M. (1966). *The Tacit Dimension*. University of Chicago Press.

[24] Geirhos, R., Jacobsen, J. H., Michaelis, C., Zemel, R., Brendel, W., Bethge, M., & Wichmann, F. A. (2020). Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11), 665–673.

[25] Chollet, F. (2019). On the measure of intelligence. *arXiv preprint arXiv:1911.01547*.