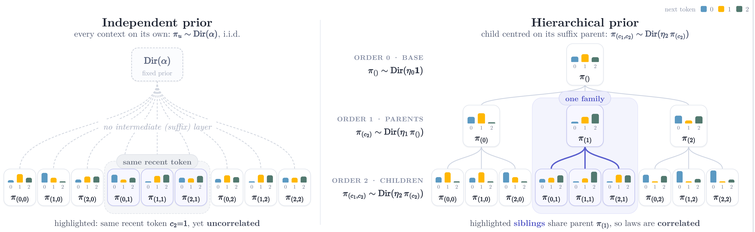
Induction Heads Interpolate N-Grams

Francesco D'Angelo* · **Oğuz Kaan Yüksel*** · Swathi Shree Narashiman · Nicolas Flammarion

Task. Order-k Markov chains with two priors



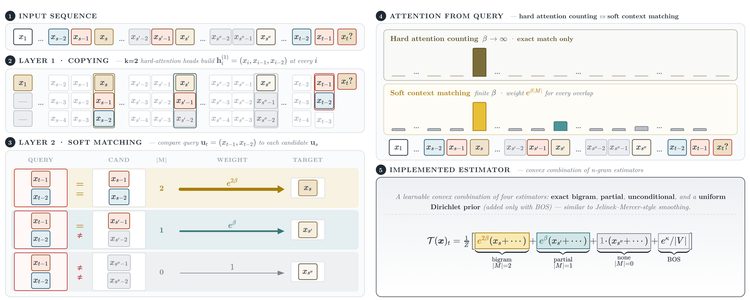
1. Sample $\boldsymbol{\pi}$ from the prior *(in-context latent transition rule)*
2. Sample $x_t \sim \mathrm{Categorical}\big(\boldsymbol{\pi}_{u_{t-1}}\big)$
   where $u_{t-1} = (x_{t-k}, \ldots, x_{t-1})$ is the order-k context
3. Predict the next token x_{T+1} from the history $x_1, \ldots, x_T$

MLE. $\hat{p}(m \mid u) = \frac{N_u(m)}{N_u}$ pure counting; zero mass on unseen contexts

Add-α. $\hat{p}(m \mid u) = \frac{\alpha_m + N_u(m)}{\sum_j \alpha_j + N_u}$ adds pseudo-counts; Bayes-optimal for the ind. prior

Notation: $N_u(m) = \#$ times m follows u; $N_u = \sum_j N_u(j)$.

Mechanism. Two-layer induction circuit



Disentangled transformer ≡ standard attention-only transformer (Nichani et al., 2024).

Pattern M contributes $\propto N_M(m)\; e^{|\boldsymbol{\beta}|_M}$ toward predicting m

κ term is exception (BOS only): an always-present context contributing pseudo-mass $e^{\kappa}/\,|V|$

$N_M(m) = \Big|\big\{s < T : \big\{i \in [k] : (u_s)_i = (u_T)_i\big\} = M,\, x_s = m\big\}\Big|$; *matches u_T* ***exactly*** *on M*

$|\boldsymbol{\beta}|_M = \sum_{i \in M} \beta_i$; *free parameters $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_k) \in \mathbb{R}^k$ (figure illustrates uniform-β case)*

Induction heads don't just count; they also regularize like classical n-gram estimators:

1. **Implement add-α smoothing**
   via the BOS token
2. **Interpolate n-grams like Jelinek–Mercer**
   via soft attention over partial matches
3. **Execute a smoothed version of Katz back-off**
   via sequence-adaptive mixture weights

Let's talk! · Paper · Code · Poster

Generalization · Learning dynamics · Interpretability

Collaborations welcome · Exploring postdoc & research scientist roles

Theory. n-gram mixture interpretation

The transformer estimator **≡** a hierarchical n-gram mixture:

1. Choose an order $i \in \{0, \ldots, k\}$ with weight $\lambda_i \propto \gamma^i K_i$
2. Choose a size-i pattern M with weight $\lambda_M \propto K_M$
3. Predict with pattern model $\hat{q}_M(m) = K_M(m)/K_M$

$\gamma = e^{\beta} - 1$; interpolation scale (assumes uniform β)

$K_M(m) = \Big|\big\{s < T : \big\{i \in [k] : (u_s)_i = (u_T)_i\big\} \supseteq M,\, x_s = m\big\}\Big|$; *matches u_T* ***at least*** *on M*

$K_M = \sum_{m \in V} K_M(m)$; total count for pattern M $\quad$ *$K_i = \sum_{|M|=i} K_M$; total count at order i*

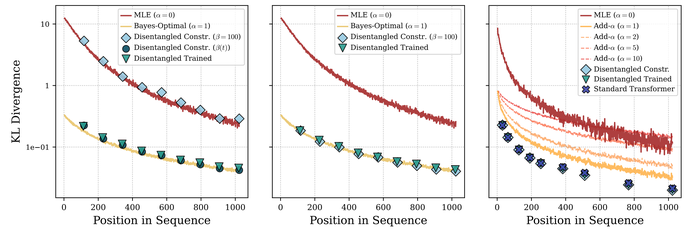
Recovers **Jelinek–Mercer smoothing** with data-dependent weights:

Weights. JM fixes λ_i globally; the transformer sets them per-sequence.

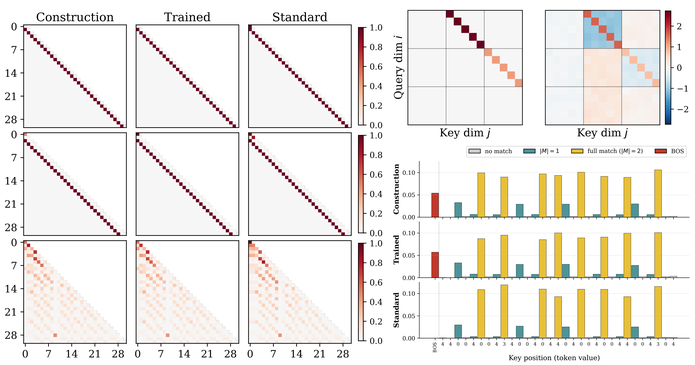
Per-order. JM uses only contiguous suffixes; the transformer averages over all size-i patterns.

Back-off. Automatic: $K_{[k]} = 0$ shifts mass to lower-order components.

Experiments. Training recovers the circuit



Performance — KL divergence under independent (no BOS / BOS) and hierarchical priors.



Mechanism (hierarchical) — Left: Layer 1 heads 1 and 2 (lags 1 and 2), and Layer 2 (matching). Right: Layer 2 weights (top) and final-position attention by context overlap (bottom).