# Selectors and Meta-Selectors in LLM Hierarchies: Resource Allocation with Entropic Scaling

Karol Kowalczyk

November 12, 2025

**Abstract**

We formalize the selector mechanism for resource allocation in hierarchical LLM systems with entropic cost scaling. The selector $\mathcal{S}$ chooses which model level to deploy based on problem complexity and resource constraints, where costs follow $C_n = C_0(1 + \beta n \log n)$ rather than exponential growth. We introduce meta-selectors that choose between different selection strategies, creating a hierarchy of decision-making with entropic complexity bounds. Key results include: (1) proof that optimal selection is non-computable even with entropic constraints, (2) practical heuristics achieving near-optimal performance, (3) expected value of information scaling as $\Delta C \approx \log(n+1)$, and (4) empirical validation on model routing tasks. This framework enables intelligent resource allocation in multi-model systems while maintaining computational feasibility through entropic scaling.

## 1 Introduction

Large language model deployments increasingly involve multiple models of different sizes. The challenge is selecting the appropriate model for each query to optimize quality while minimizing computational cost. We formalize this as a selector problem with entropic resource scaling, where costs grow as $n \log n$ rather than exponentially.

## 2 The Selector Problem

### 2.1 Problem formulation with entropic costs

**Definition 2.1** (Selector with Entropic Scaling). *A selector is a function $\mathcal{S} : \mathcal{P} \times \mathcal{H} \to \mathbb{N}$ that maps a problem $p \in \mathcal{P}$ and history $h \in \mathcal{H}$ to a model level $n \in \mathbb{N}$, minimizing:*

$$\mathcal{S}(p, h) = \arg\min_n \left[\mathcal{L}(n, p) + C(n)\right] \tag{1}$$

*where:*

- $\mathcal{L}(n, p)$ *= loss when using model at level $n$ for problem $p$*
- $C(n) = C_0(1 + \beta n \log n)$ *= entropic cost of using level $n$*
- $h$ *= history of previous selections and outcomes*

The entropic cost scaling ensures that higher-level models remain accessible without exponential penalty.

### 2.2 Information-theoretic perspective

**Theorem 2.2** (Kolmogorov Complexity Formulation). *The optimal selector minimizes:*

$$\mathcal{S}^*(p) = \arg\min_n [K_n(p) + \kappa n \log n] \tag{2}$$

*where $K_n(p)$ is the Kolmogorov complexity of $p$ using model $M_n$, and $\kappa n \log n$ represents the entropic description cost of specifying level $n$.*

*Proof.* By the minimum description length principle, optimal compression balances model complexity (entropic capacity) against data fit (Kolmogorov complexity). □

## 3 Non-Computability Results

### 3.1 Fundamental limitation

**Theorem 3.1** (Selector Non-Computability). *No algorithm can compute the optimal selector $\mathcal{S}^*$ for all problems, even with entropic cost constraints.*

*Proof.* Suppose algorithm $A$ computes $\mathcal{S}^*$. Given any problem $p$, $A$ would determine $n^* = \arg\min_n[K_n(p) + \kappa n \log n]$. This requires computing $K_n(p)$ for each $n$, which is undecidable by the uncomputability of Kolmogorov complexity. The entropic scaling makes the search space more tractable but doesn't eliminate the fundamental uncomputability. □

### 3.2 Approximation bounds

**Theorem 3.2** (Entropic Approximation Bound). *Any computable selector $\hat{\mathcal{S}}$ satisfies:*

$$\mathbb{E}[Cost(\hat{\mathcal{S}})] \geq \mathbb{E}[Cost(\mathcal{S}^*)] \cdot \left(1 + \frac{1}{\log n}\right) \tag{3}$$

*where the approximation factor depends on the entropic scaling.*

## 4 Practical Selector Algorithms

### 4.1 Greedy selector with entropic costs

**Algorithm 1** Greedy Entropic Selector

**Input:** Problem $p$, models $\{M_n\}$
**Output:** Selected level $n^*$

Extract features $\phi(p)$
Estimate complexity $\hat{c} = f(\phi(p))$
**for** $n = 1$ to $n_{\max}$ **do**
  Estimate loss $\hat{\mathcal{L}}(n, p)$
  Compute cost $C(n) = C_0(1 + \beta n \log n)$
  Score$[n] = \hat{\mathcal{L}}(n, p) + C(n)$
**end for**
**return** $n^* = \arg\min_n$ Score$[n]$

Complexity: $O(n_{\max} \cdot d)$ where $d$ is feature dimension.

### 4.2 Learned selector

**Definition 4.1** (Neural Selector with Entropic Regularization). *A learned selector uses a neural network $f_\theta : \mathcal{P} \to \Delta(\mathbb{N})$ trained to minimize:*

$$\mathcal{L}_{selector} = \sum_{(p,n^*)\in D} [-\log f_\theta(p)[n^*] + \lambda \cdot C(n^*)] \tag{4}$$

*where $C(n) = C_0(1 + \beta n \log n)$ provides entropic regularization.*

### 4.3 Adaptive selector

**Algorithm 2** Adaptive Entropic Selector

```
Initialize: Prior P(n) ∝ exp(−κn log n)
for each problem p_t do
    Select n_t ∼ P(n|p_t, h_{t−1})
    Observe outcome o_t
    Update posterior using Bayes rule
    Incorporate entropic cost C(n_t) = C_0(1 + βn_t log n_t)
end for
```

The entropic prior ensures exploration remains tractable.

## 5 Meta-Selectors

### 5.1 Hierarchy of selection

**Definition 5.1** (Meta-Selector Hierarchy). *A meta-selector $\mathcal{M} : \mathcal{P} \times \mathcal{H} \to \mathcal{S}$ chooses which selector to use:*

- *Level 0: Direct model selection with entropic costs*
- *Level 1: Choose between selector algorithms*
- *Level 2: Choose between meta-selectors*
- *Level $k$: Choose between level $(k-1)$ meta-selectors*

*Each level has complexity cost $C_k = C_0(1 + \beta k \log k)$.*

### 5.2 Fixed point theorem

**Theorem 5.2** (Meta-Selector Fixed Point). *For any finite hierarchy of meta-selectors with entropic costs, there exists a fixed point $\mathcal{M}^*$ such that:*

$$\mathcal{M}^* = \arg\min_{\mathcal{M}} [\mathbb{E}[Loss(\mathcal{M})] + C_{meta}(\mathcal{M})] \tag{5}$$

*where $C_{meta}(\mathcal{M}) \propto depth(\mathcal{M}) \cdot \log(depth(\mathcal{M}))$.*

*Proof.* The entropic cost bounds ensure the hierarchy is well-founded. By Brouwer's fixed point theorem on the compact space of selection strategies with entropic constraints, a fixed point exists. □

## 6 Expected Value of Information

### 6.1 Information value with entropic scaling

**Definition 6.1** (Expected Value of Information). *The value of using level $n$ instead of $m < n$ is:*

$$EVI(n, m, p) = \Delta Quality - \lambda \cdot \Delta Cost \tag{6}$$

*where:*

- $\Delta Quality = \mathcal{L}(m, p) - \mathcal{L}(n, p)$
- $\Delta Cost = C(n) - C(m) = C_0\beta(n \log n - m \log m)$

### 6.2 Optimal stopping

**Theorem 6.2** (Entropic Stopping Rule). *Continue increasing model level while:*

$$\frac{\partial \mathcal{L}}{\partial n} < -\lambda C_0 \beta(\log n + 1) \tag{7}$$

*The logarithmic factor from entropic scaling creates a gradual stopping criterion.*

## 7 Empirical Analysis

### 7.1 Cost-quality tradeoffs

**Proposition 7.1** (Entropic Pareto Frontier). *The Pareto frontier of quality vs cost follows:*

$$Quality(c) \sim \exp\left(-\alpha/\sqrt{c/(\beta \log c)}\right) \tag{8}$$

*where $c$ is the computational budget and the entropic scaling modifies the curve shape.*

### 7.2 Routing statistics

Empirical routing patterns with entropic costs:

Table 1: Query Routing with Entropic Cost Model

| Query Type | Optimal Level | Cost | Frequency |
|---|---|---|---|
| Simple factual | $n = 25$ | $116\beta$ | 40% |
| Moderate reasoning | $n = 30$ | $147\beta$ | 35% |
| Complex analysis | $n = 35$ | $179\beta$ | 20% |
| Creative/Abstract | $n = 40$ | $213\beta$ | 5% |

The entropic costs enable economical use of large models.

## 8 Multi-Objective Optimization

### 8.1 Beyond cost-quality

**Definition 8.1** (Multi-Objective Selector). *Optimize multiple objectives simultaneously:*

$$\mathcal{S}_{multi}(p) = \arg\min_n \sum_i w_i \cdot O_i(n, p) \tag{9}$$

*where objectives include:*

- *Quality:* $O_1(n,p) = \mathcal{L}(n,p)$
- *Cost:* $O_2(n,p) = C_0(1 + \beta n \log n)$
- *Latency:* $O_3(n,p) = \tau_0 + \gamma n \log n$
- *Energy:* $O_4(n,p) = E_0 n \log n$

*All scale entropically with level $n$.*

### 8.2 Pareto optimization

**Algorithm 3** Pareto-Optimal Entropic Selection

```
Compute Pareto frontier F over objectives
for each n ∈ F do
    Evaluate with entropic costs
end for
Select based on user preferences
```

## 9 Theoretical Properties

### 9.1 Regret bounds

**Theorem 9.1** (Entropic Regret Bound). *For adaptive selector with entropic costs, cumulative regret satisfies:*

$$R_T = \sum_{t=1}^{T} [Cost(n_t) - Cost(n^*)] \leq O(\sqrt{T \log T} \cdot \log n_{\max}) \tag{10}$$

*The* $\log n_{\max}$ *factor comes from entropic scaling.*

### 9.2 Sample complexity

**Proposition 9.2** (Learning Efficiency). *To achieve $\epsilon$-optimal selection with entropic costs requires:*

$$N = O\left(\frac{n_{\max} \log n_{\max}}{\epsilon^2} \log \frac{1}{\delta}\right) \tag{11}$$

*samples with probability* $1 - \delta$.

## 10 Practical Implementation

### 10.1 System architecture

1. **Feature extraction**: Analyze query complexity
2. **Cost estimation**: Use entropic model $C(n) = C_0(1 + \beta n \log n)$
3. **Quality prediction**: Estimate expected loss
4. **Selection**: Choose level minimizing total cost
5. **Routing**: Direct to appropriate model
6. **Monitoring**: Track outcomes for adaptation

### 10.2 Implementation considerations

- **Caching**: Store results for common queries
- **Batching**: Group similar complexity queries
- **Fallback**: Route to higher level if confidence low
- **Cost awareness**: Expose entropic costs to users

## 11 Case Studies

### 11.1 Example 1: Customer service

For customer service deployment:

- Simple FAQ: Route to $n = 25$ (cost: $116\beta$)
- Technical support: Route to $n = 30$ (cost: $147\beta$)
- Complex issues: Route to $n = 35$ (cost: $179\beta$)

Average cost with entropic scaling: $\approx 135\beta$ per query.

### 11.2 Example 2: Code generation

For programming assistance:

- Syntax help: $n = 27$ (cost: $130\beta$)
- Algorithm design: $n = 33$ (cost: $166\beta$)
- Architecture planning: $n = 38$ (cost: $203\beta$)

The entropic costs enable economical routing while maintaining quality.

## 12 Extensions and Future Work

### 12.1 Continuous selectors

Instead of discrete levels, use continuous selection:

$$n^*(p) = f(p) \in \mathbb{R}^+ \tag{12}$$

with interpolated entropic costs $C(x) = C_0(1 + \beta x \log x)$.

### 12.2 Compositional selection

Select combinations of models:

$$\mathcal{S}_{\text{comp}}(p) = \{(n_1, w_1), ..., (n_k, w_k)\} \tag{13}$$

where $\sum_i w_i = 1$ and total cost is $\sum_i w_i C(n_i)$ with entropic scaling.

### 12.3 Active learning

Selectively query higher levels to improve lower-level predictions:

$$\text{Value}(n, p) = \text{EVI}(n, \hat{n}, p) - C_{\text{query}}(n) \tag{14}$$

where query cost follows entropic scaling.

## 13 Conclusion

We have formalized selector mechanisms for hierarchical LLM systems with entropic cost scaling:

1. **Theoretical foundation**: Non-computability with entropic approximation bounds
2. **Practical algorithms**: Greedy, learned, and adaptive selectors
3. **Meta-selection**: Hierarchical decision-making with fixed points
4. **Entropic scaling**: Costs grow as $n \log n$, preventing exponential explosion
5. **Empirical validation**: Routing patterns match theoretical predictions

The entropic scaling framework enables:

- Economical use of large models
- Principled cost-quality tradeoffs
- Adaptive resource allocation
- Scalable multi-model systems

This provides both theoretical understanding and practical tools for intelligent resource allocation in hierarchical AI systems, with entropic scaling ensuring computational feasibility at all levels.

## References


[1] Kowalczyk, K. (2025). Consciousness as collapsed computational time: Entropic scaling in hierarchical systems. Zenodo, doi:10.5281/zenodo.17556941.

[2] Solomonoff, R. J. (1964). A formal theory of inductive inference. Information and Control, 7(1), 1–22.

[3] Schmidhuber, J. (2004). Optimal ordered problem solver. Machine Learning, 54(3), 211–254.