From Counted Context to Recurrent Attribution

An Explainer for MCP Grafs 15, 21, 22, 23, and 24

Claude and MJC

March 12, 2026

Abstract

The MCP now contains a compact thread that starts with context selection and ends with recurrent attribution. Prediction should be based on *counted contexts*, but the best context is not necessarily the longest contiguous window. A small number of well-chosen offsets can outperform a much longer block of adjacent bytes because distant positions may contribute *complementary* information rather than redundant information. Once those informative offsets have been identified in the UM, the next question is how an RNN carries the same information in hidden state. This note explains five MCP grafs: the coarse/fine code split, the global prior from log-count support, the greedy offset rule, the exact conditional distribution from counted context, and sparse differentiation for recurrent attribution.

1 The Five Grafs Verbatim

The core equations are:

$$\mathrm{bpc}_{ES} = H(ES_{t+1} \mid ES_t) + \mathbb{E}[\log_2 |ES_{t+1}|] \tag{1}$$

$$A(b) = \log_2 c(b) \tag{2}$$

$$P(b) = \frac{2^{A(b)}}{\sum_{b'} 2^{A(b')}} = \frac{c(b)}{N} \tag{3}$$

$$o_{k+1} = \arg\min_{o} H(Y \mid X_{o_1}, \ldots, X_{o_k}, X_o) \tag{4}$$

$$P(y \mid x_1, \ldots, x_k) = \frac{\mathrm{count}(x_1, \ldots, x_k, y)}{\mathrm{count}(x_1, \ldots, x_k)} \tag{5}$$

$$g_t = \nabla_{h_t} \log P(y \mid h_t) = W_y^T (e_y - P) \tag{6}$$

2 Counted Context

Definition 1 (Counted conditional)**.** *Given a context $(x_1, \ldots, x_k)$, the model predicts the next symbol by looking at all past occurrences of that context and counting what followed:*

$$P(y \mid x_1, \ldots, x_k) = \frac{\mathrm{count}(x_1, \ldots, x_k, y)}{\mathrm{count}(x_1, \ldots, x_k)}.$$

This is the simplest exact context model. There is no hidden machinery: prediction is just a conditional frequency table.

Example 1 (Two contexts)**.** *Suppose the context* `th` *occurred 100 times, followed by:*

$$\texttt{e} : 75, \quad \texttt{a} : 12, \quad \texttt{i} : 8, \quad \texttt{o} : 4, \quad \texttt{u} : 1.$$

Then

$$P(\texttt{e} \mid \texttt{th}) = 0.75,$$

and similarly for the other letters. The context does not identify the next letter perfectly, but it narrows the distribution sharply.

3 The Global Prior

Before using any context at all, the model still has a prior: the marginal output frequencies. If byte b occurs $c(b)$ times in N positions, then its log-count support is

$$A(b) = \log_2 c(b).$$

Exponentiating and normalizing recovers the marginal:

$$P(b) = \frac{2^{A(b)}}{\sum_{b'} 2^{A(b')}} = \frac{c(b)}{N}.$$

So the order-0 model is already a valid probability model. Context refines this prior; it does not replace it.

4 Coarse and Fine Coding

The coarse/fine equation

$$\mathrm{bpc}_{ES} = H(ES_{t+1} \mid ES_t) + \mathbb{E}[\log_2 |ES_{t+1}|]$$

has a simple meaning.

First pay to predict the next coarse class. Then pay to identify the exact symbol inside that class.

This is the same basic split we saw in the quotient explainer:

1. coarse uncertainty about *which bucket*;
2. fine uncertainty about *which member of the bucket*.

The counted conditional model generalizes this. Instead of only using a coarse event space, we can use any context we like. The question becomes: which context should we keep?

5 Why More Nearby Context Can Be Wasteful

It is tempting to think that the best context is always the longest contiguous suffix: the previous byte, then the previous two bytes, then the previous three, and so on.

But adjacent positions are often redundant. If offset 1 already tells us most of what offset 2 would tell us, then adding offset 2 may not buy much. A farther offset can be better if it contributes information that the closer offsets do not already contain.

That is why skip-patterns matter. They let us choose *which* past positions to keep, not just how many.

6 Greedy Offset Selection

The MCP rule is:

$$o_{k+1} = \arg\min_{o} H(Y \mid X_{o_1}, \ldots, X_{o_k}, X_o).$$

Read this operationally:

1. start with the currently selected offsets;
2. test each candidate new offset;
3. compute the remaining uncertainty about Y;
4. pick the offset that reduces that uncertainty the most.

Proposition 1. *The greedy rule prefers* complementary *offsets over merely nearby ones. A candidate offset is valuable when it lowers conditional entropy after the current offsets are already known.*

So the rule does not ask "which offset is strongest by itself?" It asks "which offset adds the most new information given what we already know?"

7 A Toy Illustration

Suppose the previous byte X_1 already tells us whether we are inside a word. Then offset 2 may add little, because it tends to be correlated with offset 1.

But offset 8 might often cross a tag boundary, line structure, or some other longer-range regularity. Even if offset 8 is weaker in isolation, it may reduce

$$H(Y \mid X_1, X_8)$$

more than offset 2 reduces

$$H(Y \mid X_1, X_2).$$

That is the central skip-pattern phenomenon: a farther clue can be more valuable because it is less redundant.

8 Why The Right Context Beats More Context

We can now state the main point clearly:

1. Prediction comes from counted conditionals.
2. The order-0 prior is just counted marginals.
3. Coarse/fine coding explains why context can be decomposed into bucket choice plus residual choice.
4. The best added context is the one that most reduces conditional entropy, not necessarily the nearest byte.

So a short skip-context can beat a long contiguous context. The issue is not raw context length. The issue is whether each added piece of context contributes independent predictive information.

9 Relation to Hidden State

The counted-context UM makes every useful offset explicit. The RNN does not: it compresses those useful offsets into hidden state.

That is why the skip-pattern story is a good explainer for the hidden state as well. The hidden layer is one way of carrying the right clues forward. The UM shows those clues directly by counting.

10 From Explicit Context to Recurrent Attribution

Graf 24 asks a new question. Once the RNN has made a prediction, how do we identify which past positions actually carried the signal?

The first step is the local gradient of the correct prediction with respect to the hidden state:

$$g_t = \nabla_{h_t} \log P(y \mid h_t) = W_y^T (e_y - P).$$

This is the analogue, inside the RNN, of asking which directions in state space would most increase the probability of the correct next symbol.

To trace that signal backwards, we use the recurrent Jacobian

$$J_t = \frac{\partial h_t}{\partial h_{t-1}} = \mathrm{diag}(1 - h_t^2)\, W_h,$$

and transport the gradient from time t back to time $t-k$:

$$g_{t \to t-k} = \left(\prod_{s=t-k+1}^{t} J_s \right)^T g_t.$$

This says: if the prediction gradient is the "question" at time t, then the transported gradient is the same question rewritten in the coordinates of an earlier hidden state.

Finally, to score a concrete past input at offset k, graf 24 uses

$$\mathrm{attr}(t, k) = W_x[:, x_{t-k}]^T g_{t \to t-k}.$$

This is the attribution of the specific symbol at position $t-k$ to the prediction made at position t.

11 Why Graf 24 Belongs After Graf 23

Graf 23 gives the exact counted conditional law:

$$P(y \mid x_1, \ldots, x_k) = \frac{\mathrm{count}(x_1, \ldots, x_k, y)}{\mathrm{count}(x_1, \ldots, x_k)}.$$

That equation tells us which contexts matter *in the explicit UM.*

Graf 24 then asks how the same predictive dependence appears *in the RNN.* The answer is not another count table. The answer is a backward transport calculation that measures which earlier inputs moved the hidden state in the direction that helped the current prediction.

So the two grafs form a bridge:

1. graf 23 identifies predictive structure explicitly as counted context;
2. graf 24 traces the corresponding predictive signal implicitly through recurrent state.

The point of the bridge is not that counts and gradients are the same object. The point is that they answer the same question in two different representations: which earlier observations are carrying the information that determines the next symbol?

12 Conclusion

The counted-context thread in MCP says:

1. start from counts;
2. recover the prior from log-count support;
3. refine the prior with conditional counts;
4. choose additional offsets by minimizing the remaining entropy.
5. trace the chosen predictive signal through recurrent state by backward transport and input attribution.

The result is a precise bridge from explicit context to hidden-state attribution. The best context is the one that adds new information, and graf 24 shows how to locate that information once it has been packed into recurrence.

References

[1] Claude and MJC. *MCP.* Working manuscript, March 2026.

[2] Claude and MJC. *The Pattern-Chain UM.* Hutter archive, 8 February 2026.

[3] Claude and MJC. *Pattern Priors and Skip-Patterns.* Hutter archive, 8 February 2026.

[4] Claude and MJC. *Sparse Differentiation.* Hutter archive, 9 February 2026.