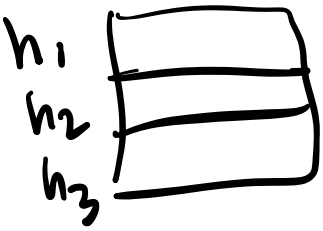
self-attn:

$$H = \text{softmax}\left(\text{mask}\left(QK^T\right)\right)V$$

W_q W_k W_v



$q_1 = W_q c_1$

params: $\{W_q, W_k, W_v, W_o, E\}$

Word order / position info

$p_1 + c_1$ favorite

$p_2 + c_2$ LLM

$p_3 + c_3$ is

learn a separate embedding for each position → absolute pos. emb additive

No positional encoding (NoPE):

↳ how can a self-attn model still learn position info

↳ masked self-attn

↳ mask leaks position

ex: $W_q = 0$



favorite LLM is

value vector <sos> : <1, 0 ...>

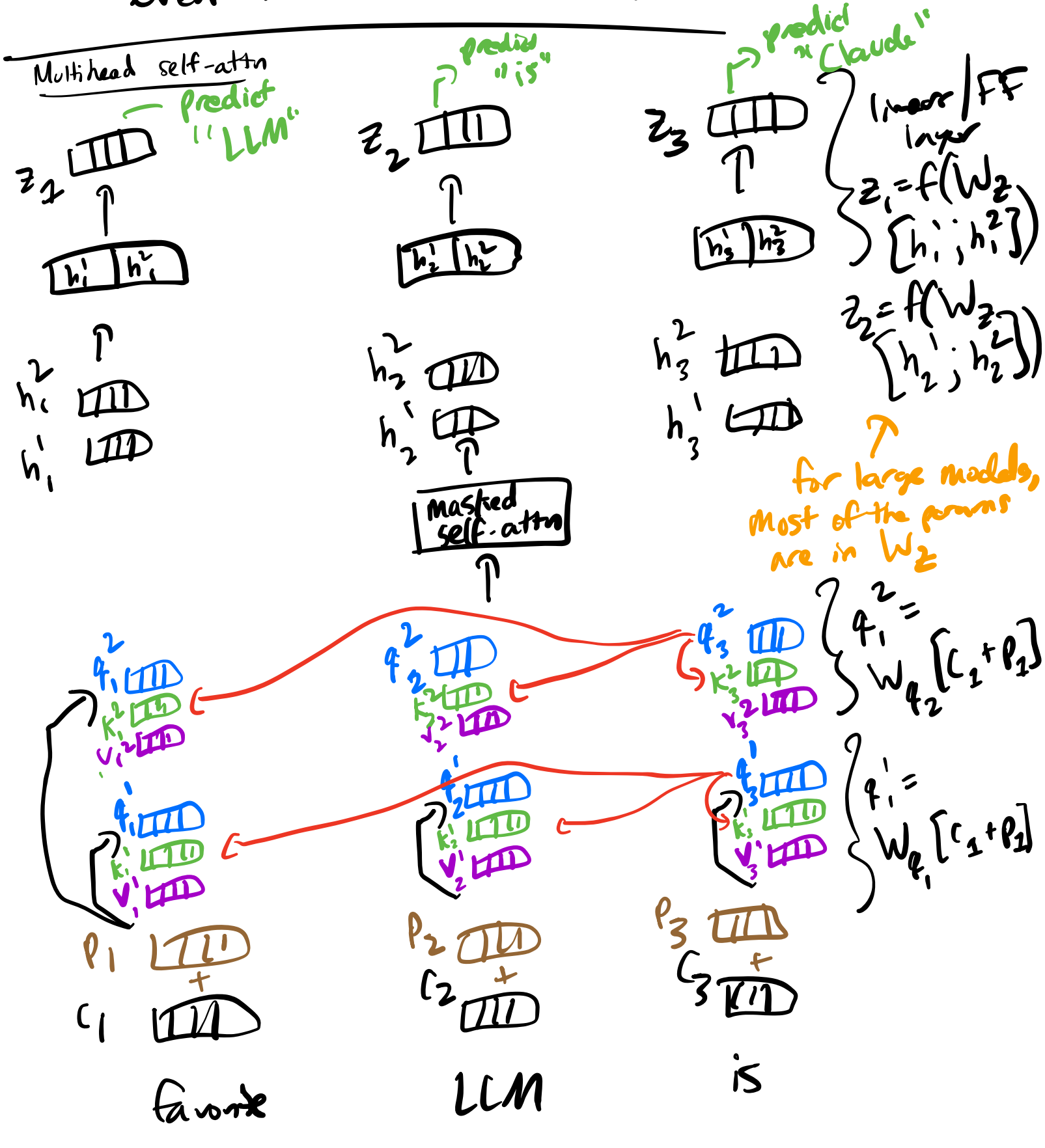
... every other word : <0, 1 ...>

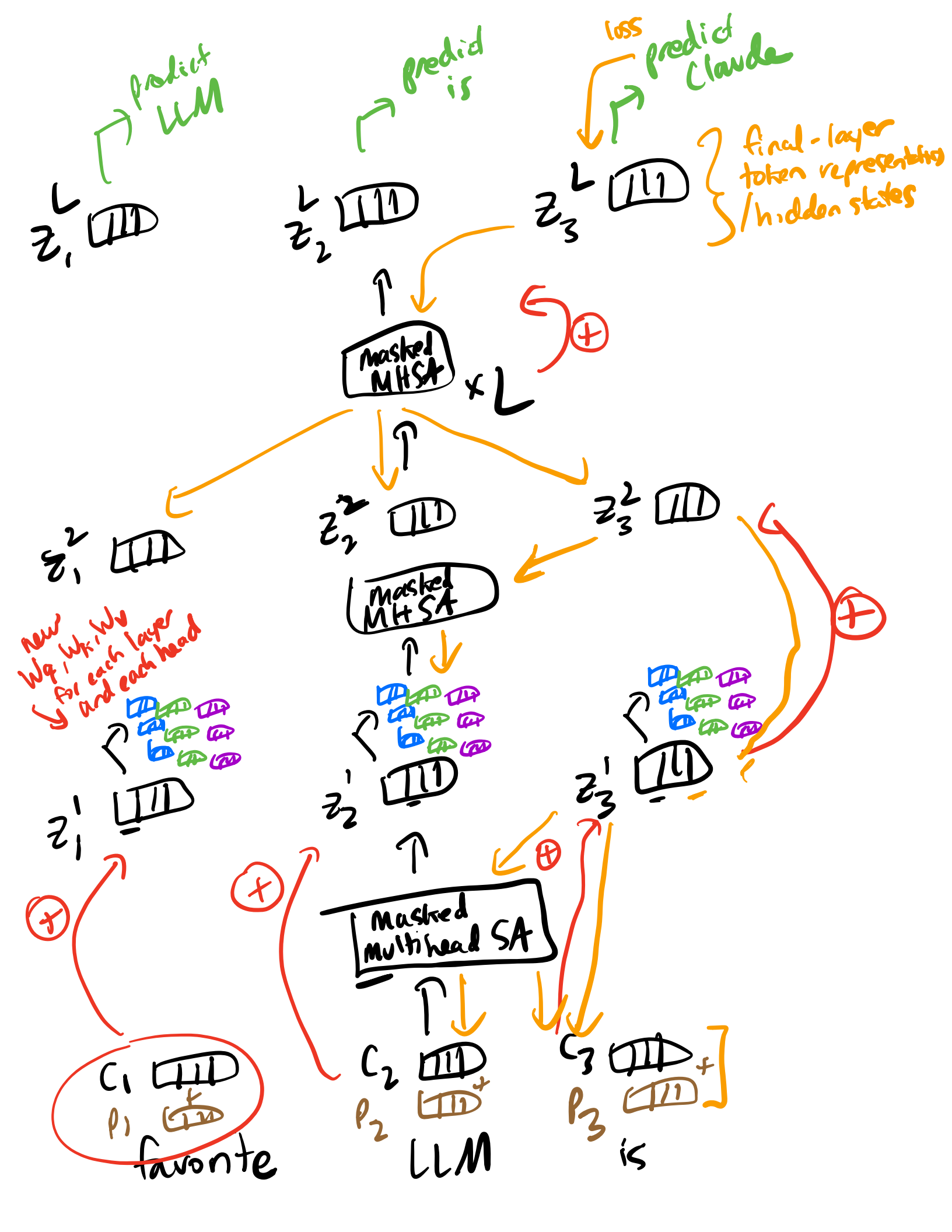
hidden state h_t is just a weighted average of value vectors

$$h_3 = \frac{1}{3}\langle 1,0\rangle + \frac{1}{3}\langle 0,1\rangle + \frac{1}{3}\langle 0,1\rangle$$

$$= \left\langle \frac{1}{3}, \frac{2}{3} \right\rangle$$

Position encodings offer useful inductive bias to the model, even if it's not strictly necessary




predict LLM
predict is
loss
predict Claude
final-layer token representations / hidden states
masked MHSA
× L
new W_q, W_k, W_v for each layer and each head
masked MHSA
masked multihead SA
favorite
LLM
is

test-time

↳ predicting one word at a time

↳ input: user-provided prefix (prompt)

task: decode response based on the prefix

2 phases:

↳ prefill: in parallel, compute keys values of the input prefix

↳ form KV cache

↳ decoding: generate one token per step