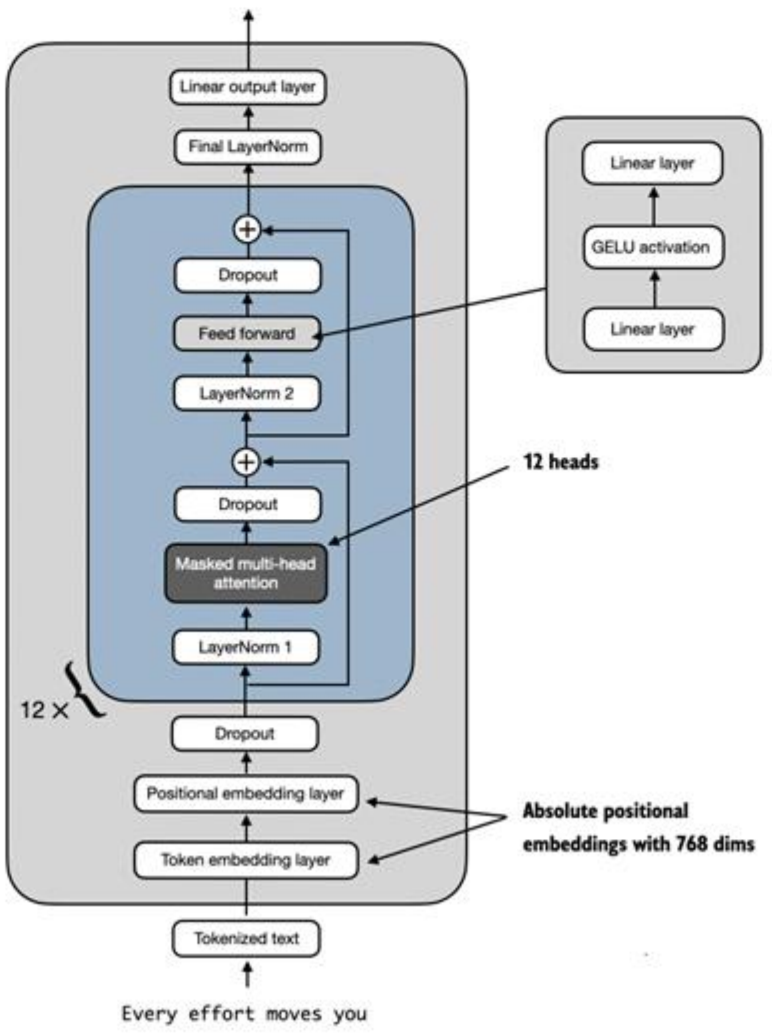
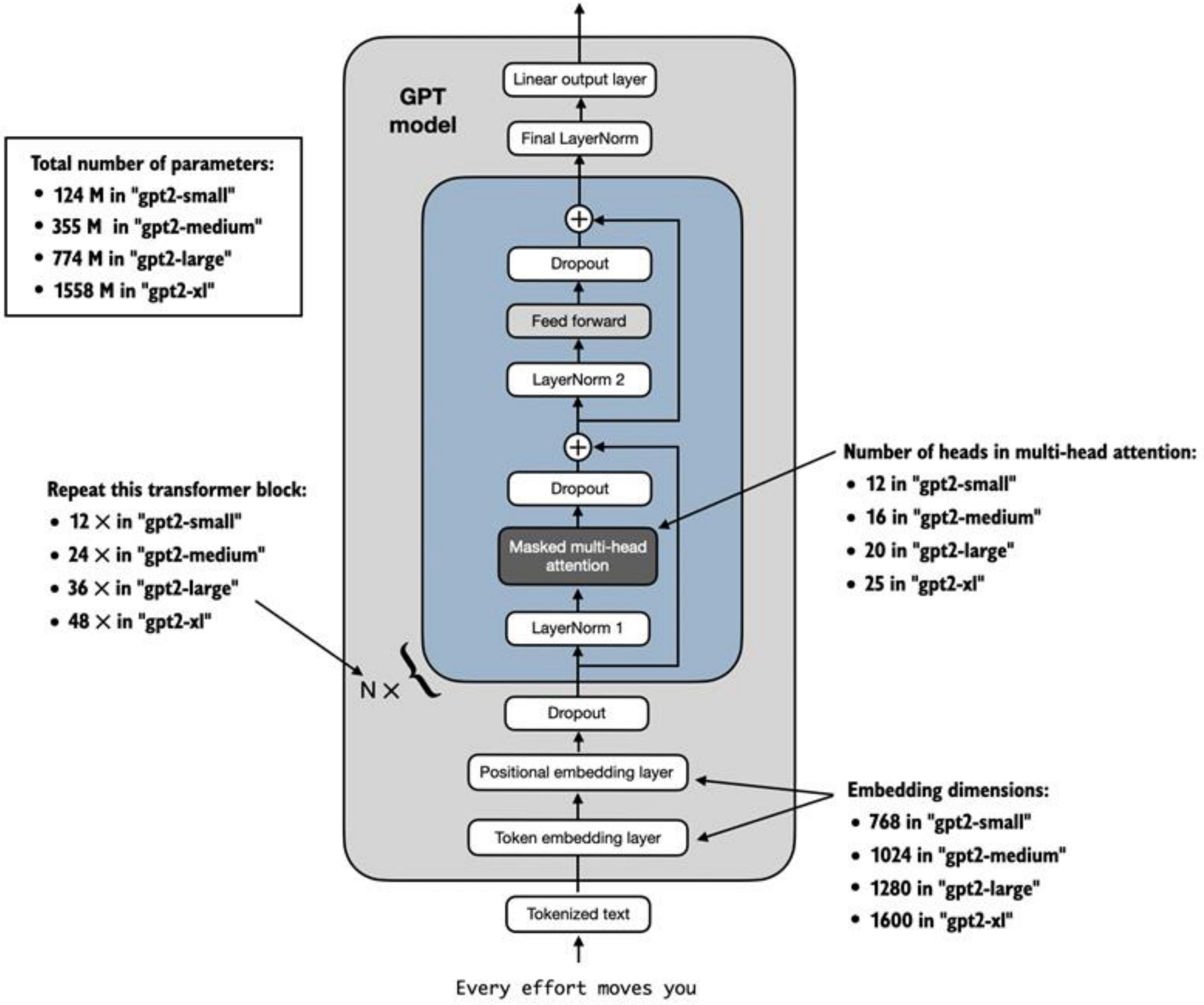
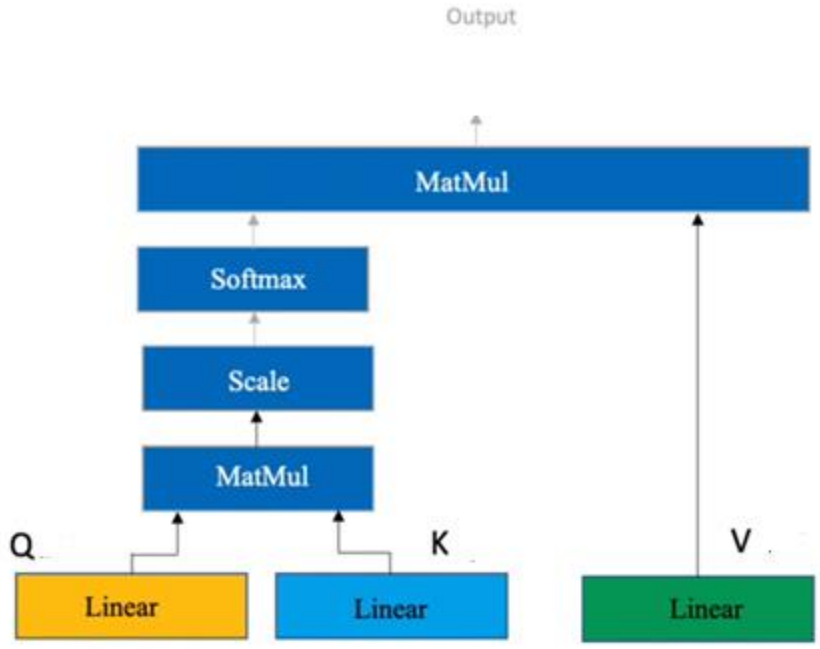
GPT-2 (small) Model



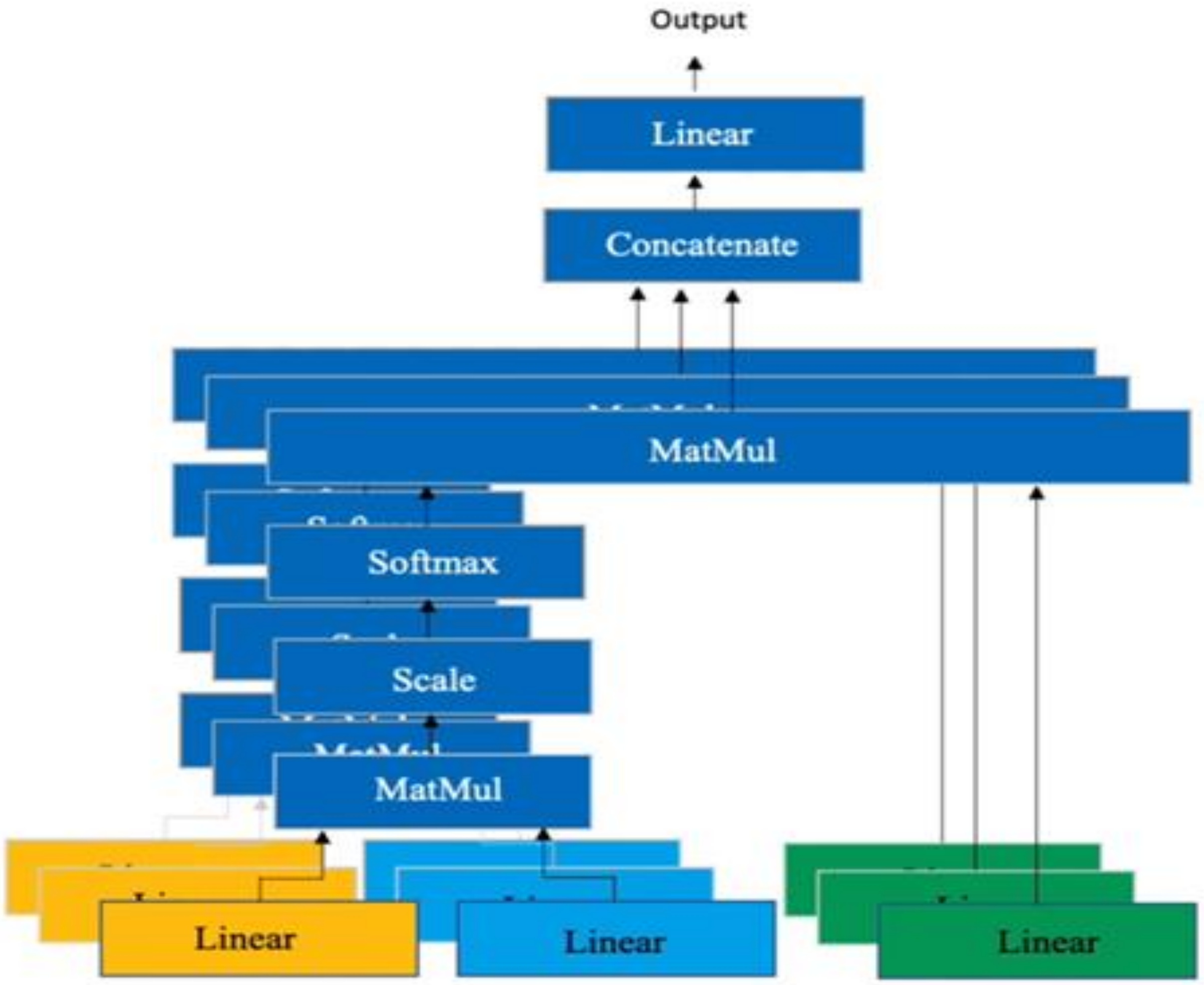

GPT model
Linear output layer
Final LayerNorm
Dropout
Feed forward
LayerNorm 2
Dropout
Masked multi-head attention
LayerNorm 1
N ×
Dropout
Positional embedding layer
Token embedding layer
Tokenized text
Every effort moves you
Total number of parameters:
• 124 M in "gpt2-small"
• 355 M in "gpt2-medium"
• 774 M in "gpt2-large"
• 1558 M in "gpt2-xl"
Repeat this transformer block:
• 12 × in "gpt2-small"
• 24 × in "gpt2-medium"
• 36 × in "gpt2-large"
• 48 × in "gpt2-xl"
Number of heads in multi-head attention:
• 12 in "gpt2-small"
• 16 in "gpt2-medium"
• 20 in "gpt2-large"
• 25 in "gpt2-xl"
Embedding dimensions:
• 768 in "gpt2-small"
• 1024 in "gpt2-medium"
• 1280 in "gpt2-large"
• 1600 in "gpt2-xl"

Single-head Attention



$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{Q.K^T}{\sqrt{d_k}}\right).V$$

Multi-head Attention



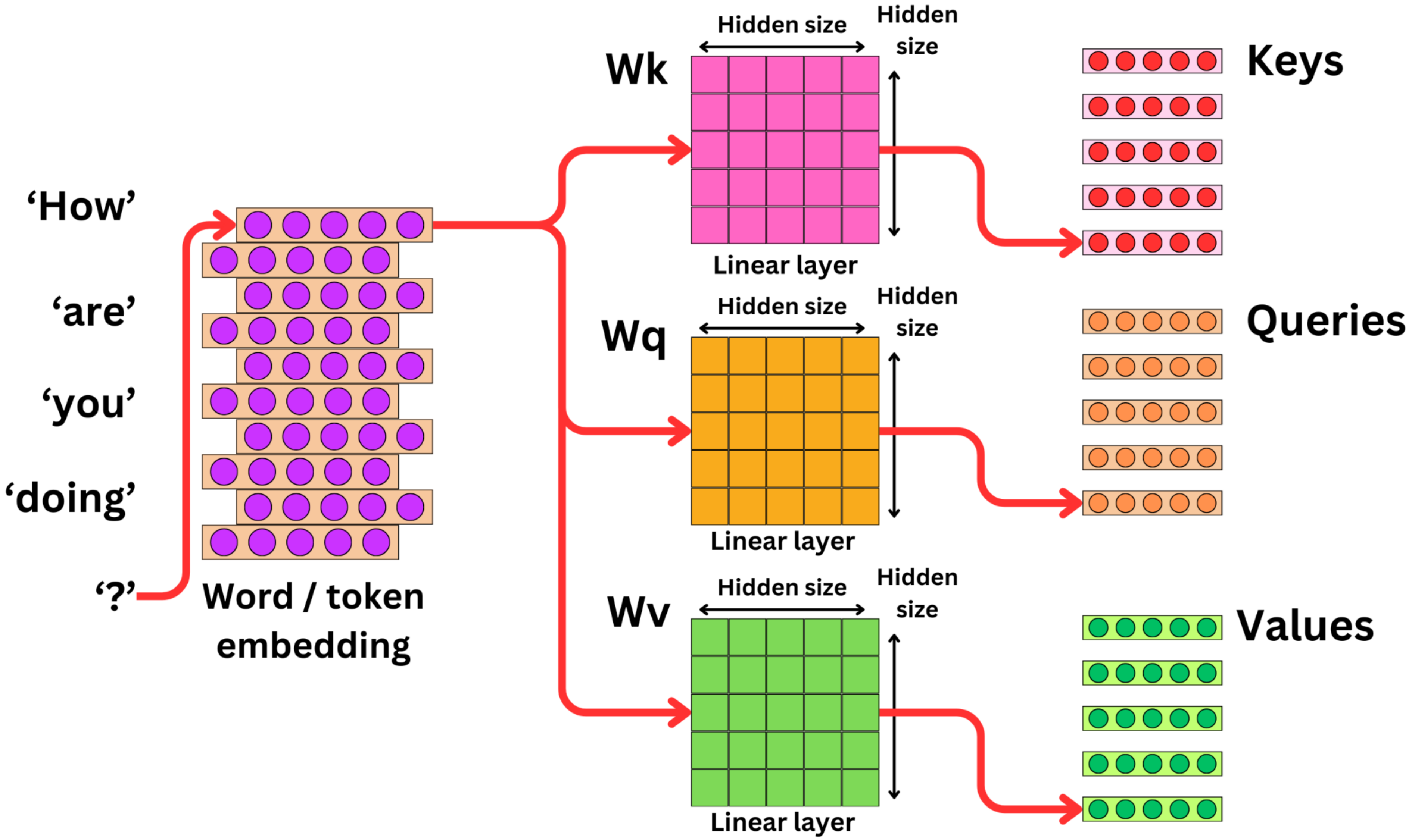
Attention

“The bank is steep, so it’s dangerous to stand near it.”

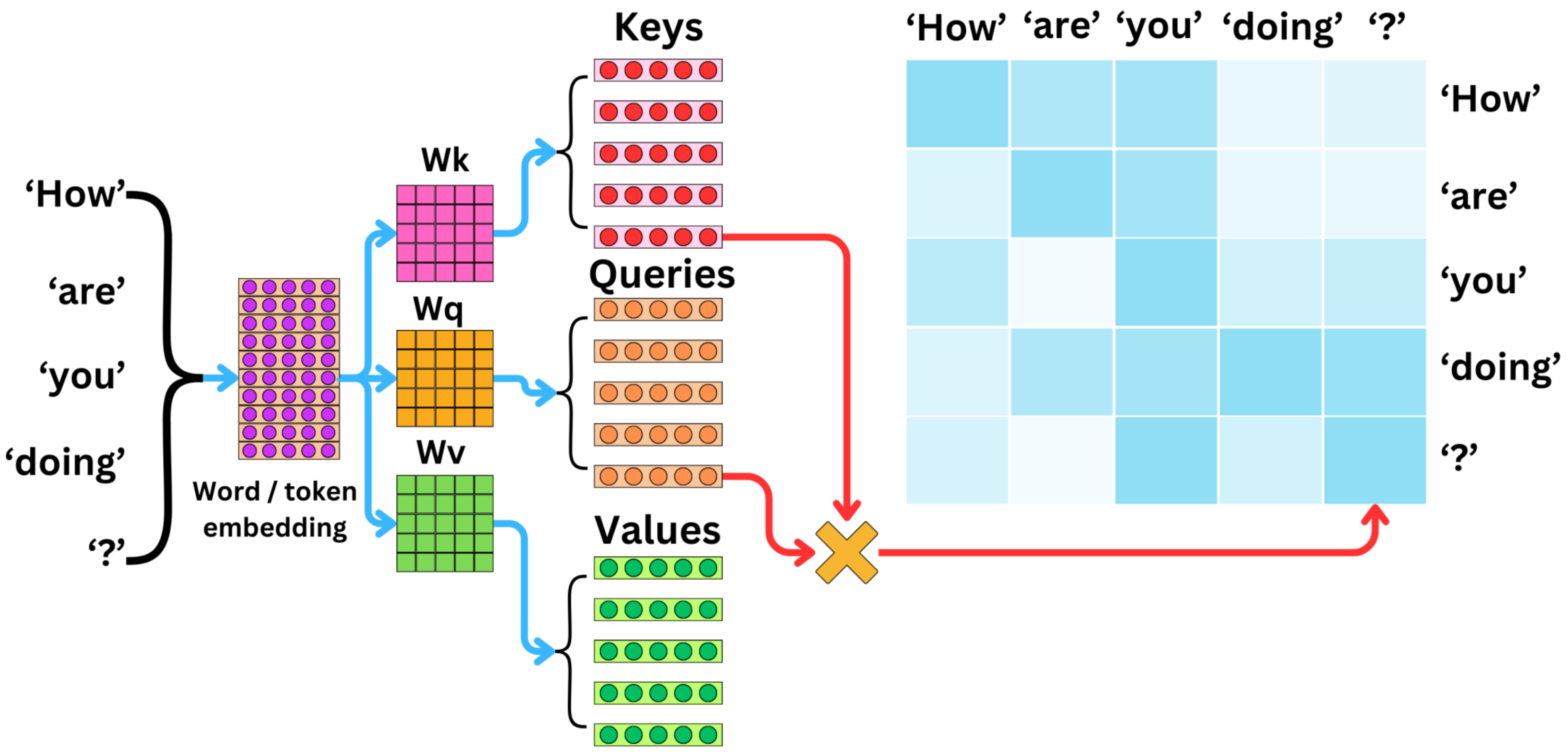
- Query (*“it”*): “What does *’it’* refer to?”
- Keys (*“bank,” “steep,” “dangerous”*): Highlight candidates for reference.
- Values: Encode the meaning of each candidate.

The model computes high attention weights between the query (*“it”*) and keys (*“bank,” “steep”*), then aggregates their values to infer *“it”* refers to the riverbank.

Single head Attention

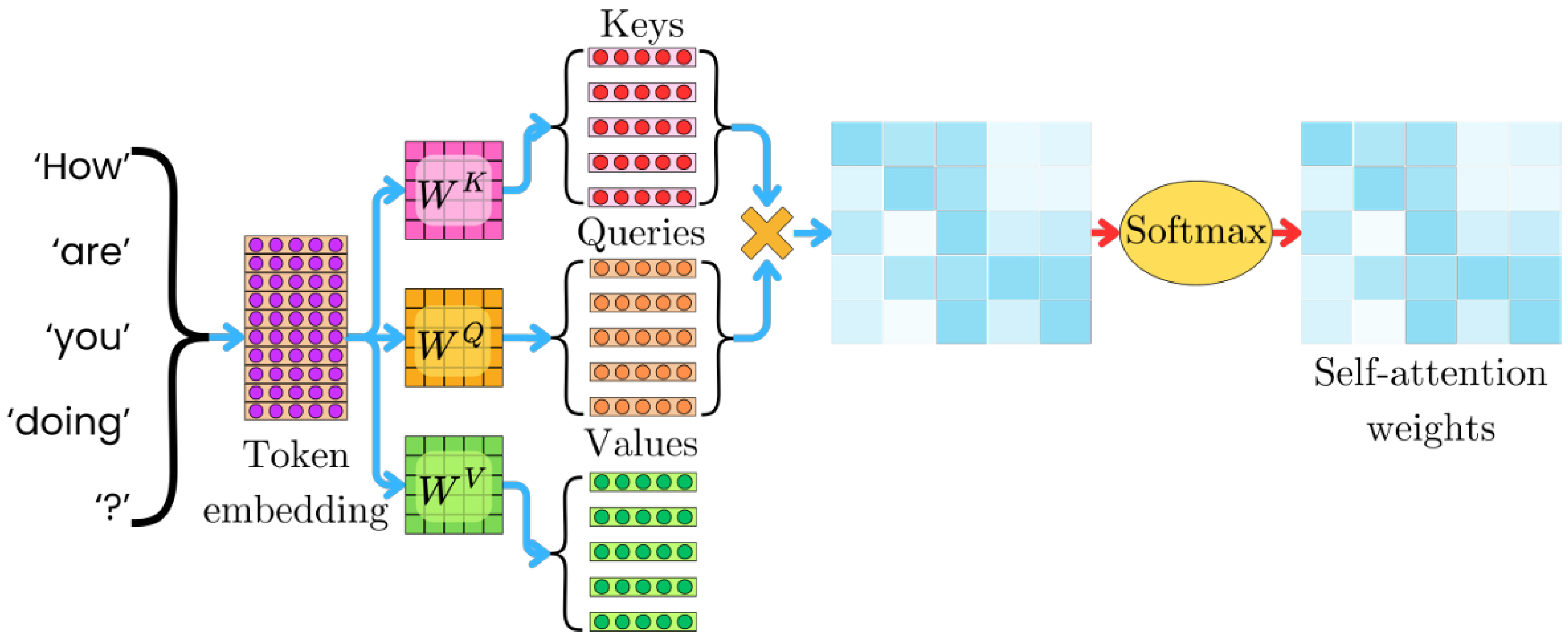


Single head Attention



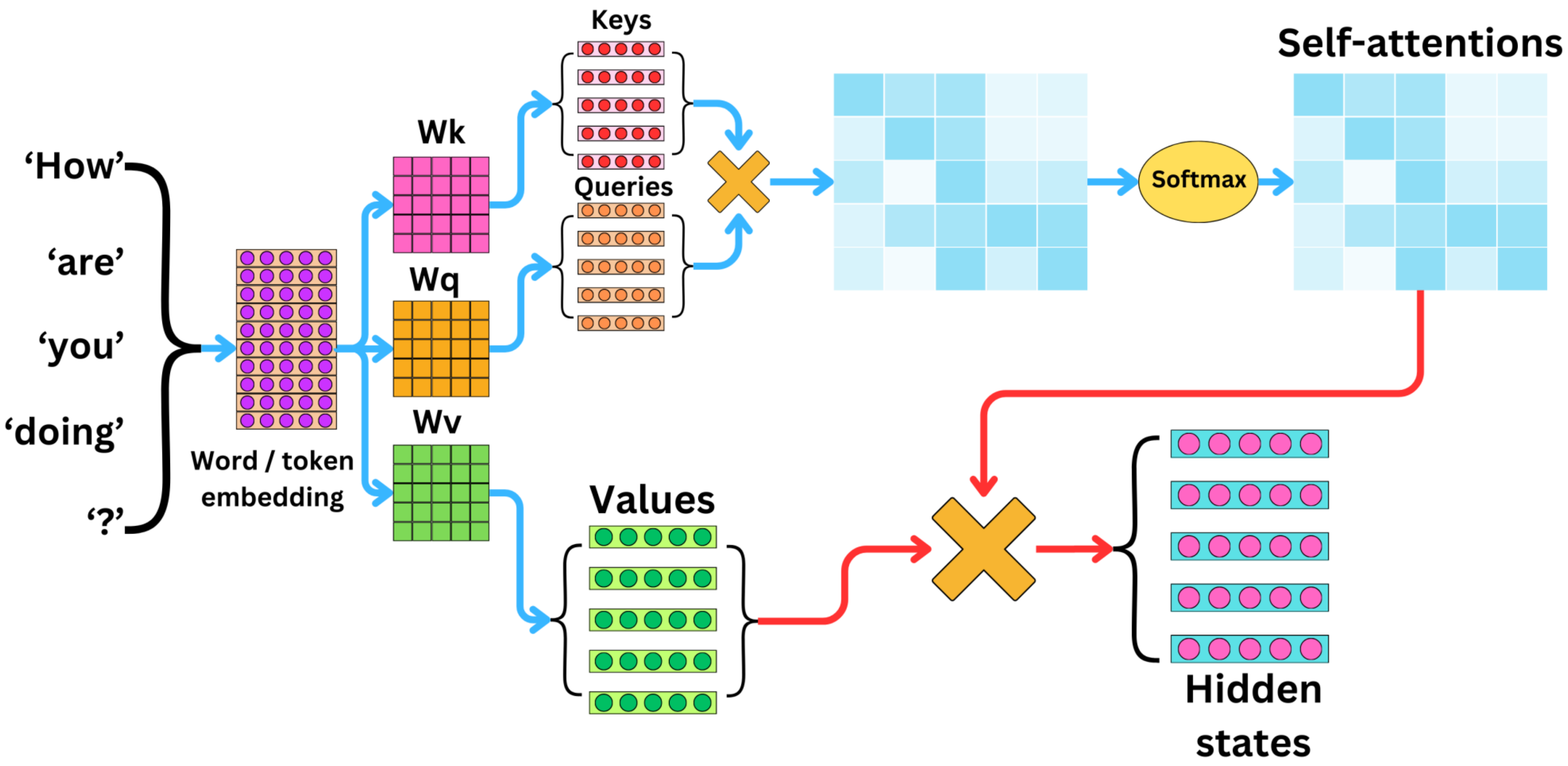
$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{Q.K^T}{\sqrt{d_k}}\right).V$$

Single head Attention



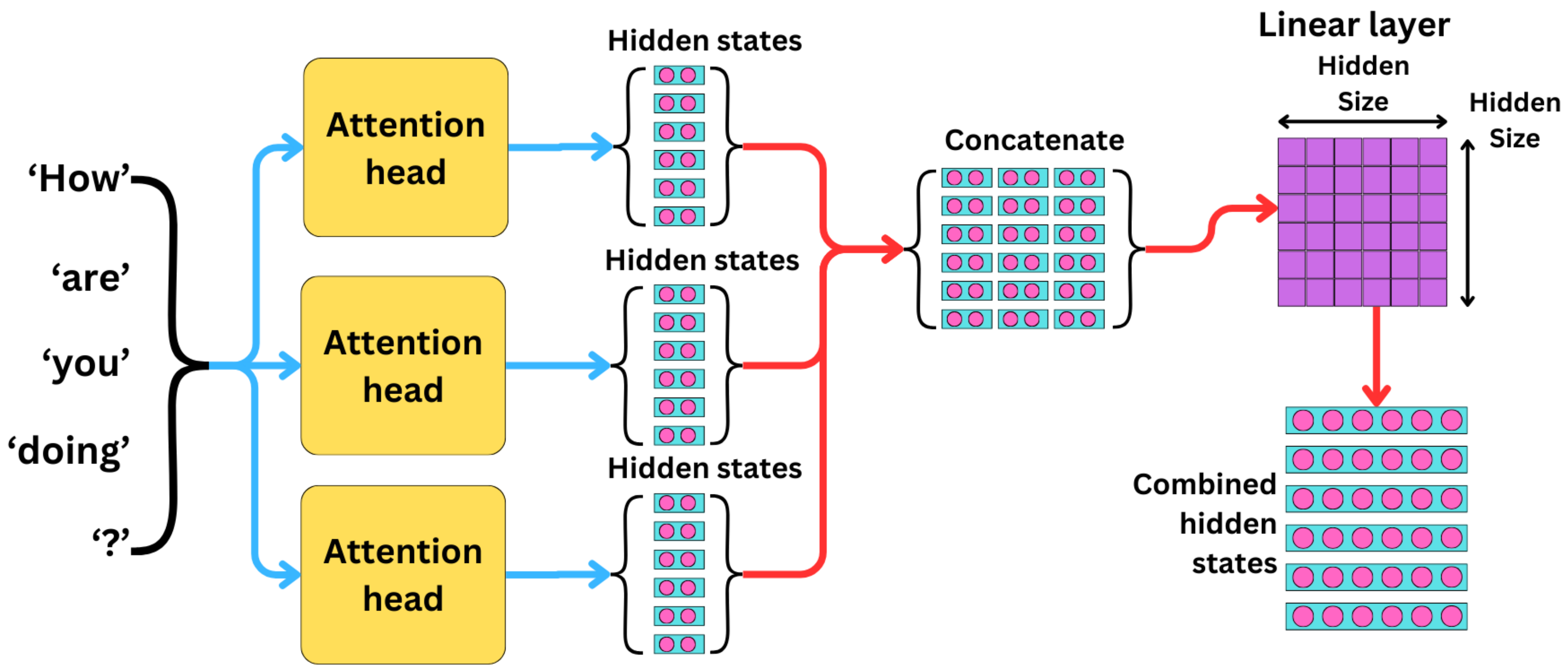
$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{Q.K^T}{\sqrt{d_k}}\right).V$$

Single head Attention

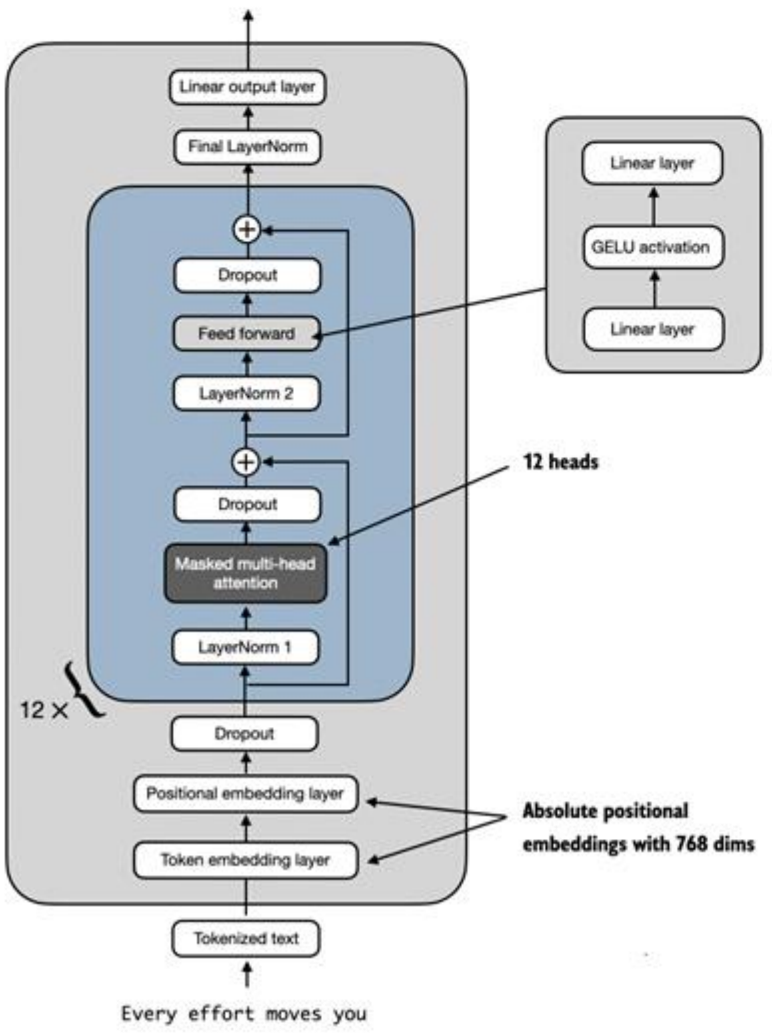


$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{Q.K^T}{\sqrt{d_k}}\right).V$$

Multi head Attention

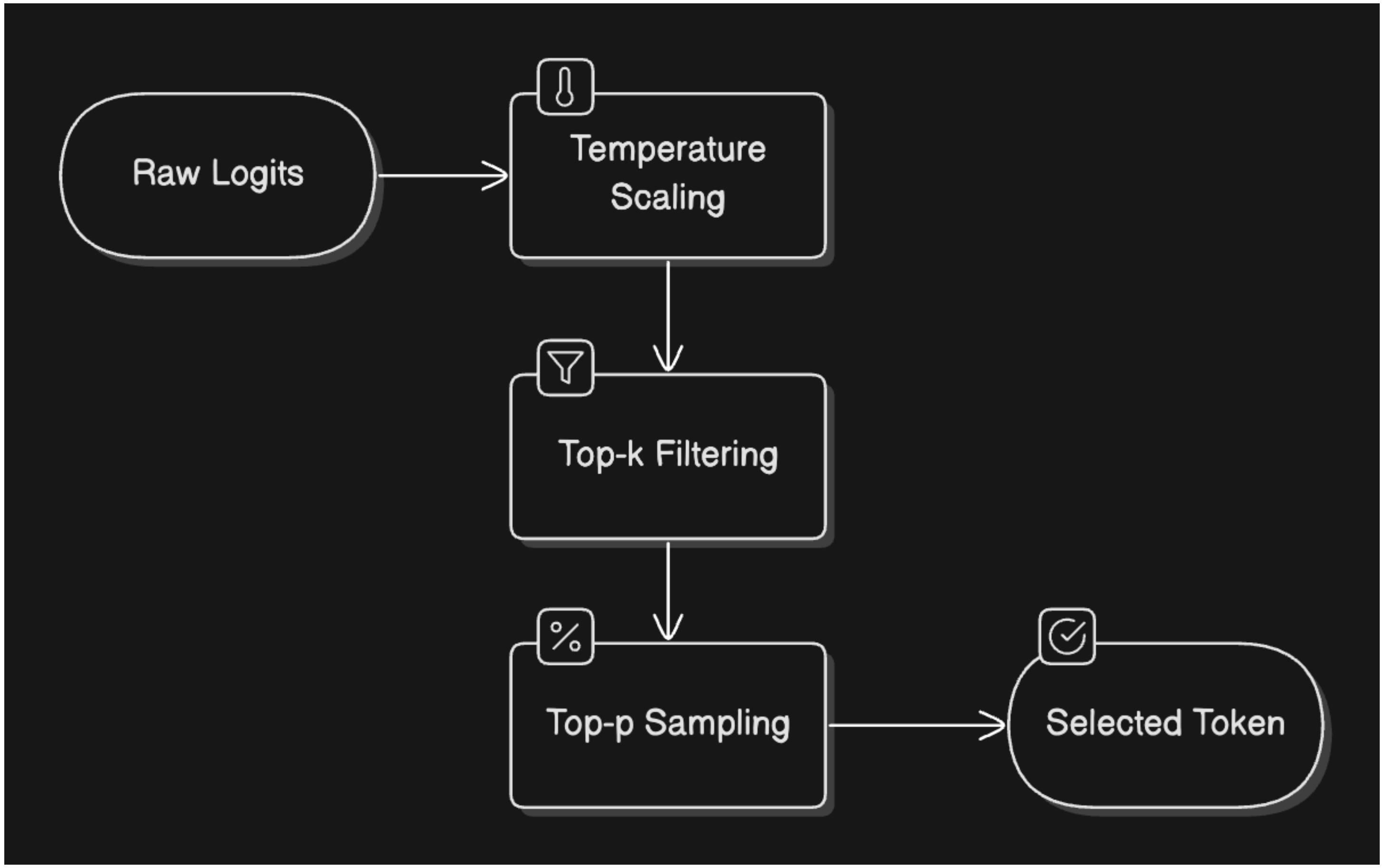


GPT-2 (small) Model



Aspects	GPT-1	GPT-2	GPT-3
No of parameters	117 M	1.5 B	175 B
Training dataset size	Approx. 5 GB	40 GB	570 GB
Sequence length	512	1024	2048
Vocabulary size	40,000	50,257	50,257
Embedding dimension	768	1600	12,288
No of attention heads	12	25	96
Hidden layer size	3072	6400	49,152
No fo decoder layers	12	48	96

Generation Control



Generation Control

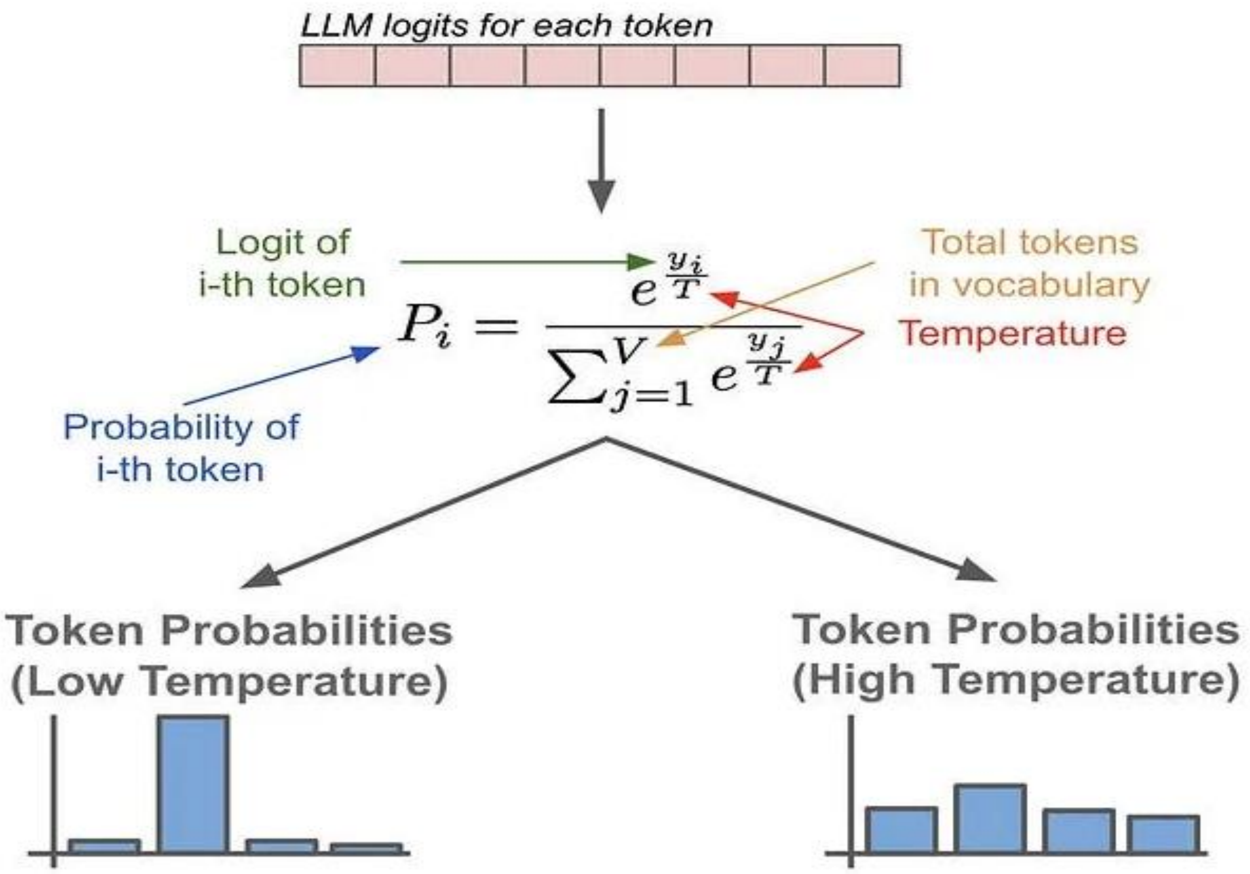
- **Temperature Control**: Like a creativity dial - higher settings (>1.0) make choices more random and creative, lower settings (<1.0) make them more focused and deterministic
- **Top-p (Nucleus) Sampling**: Instead of considering all possible words, we only look at the most likely ones that add up to our chosen probability threshold (e.g., top 90%)
- **Top-k Filtering**: An alternative approach where we only consider the k most likely next words

Temperature in LLM APIs

`temperature` number Optional Defaults to 1

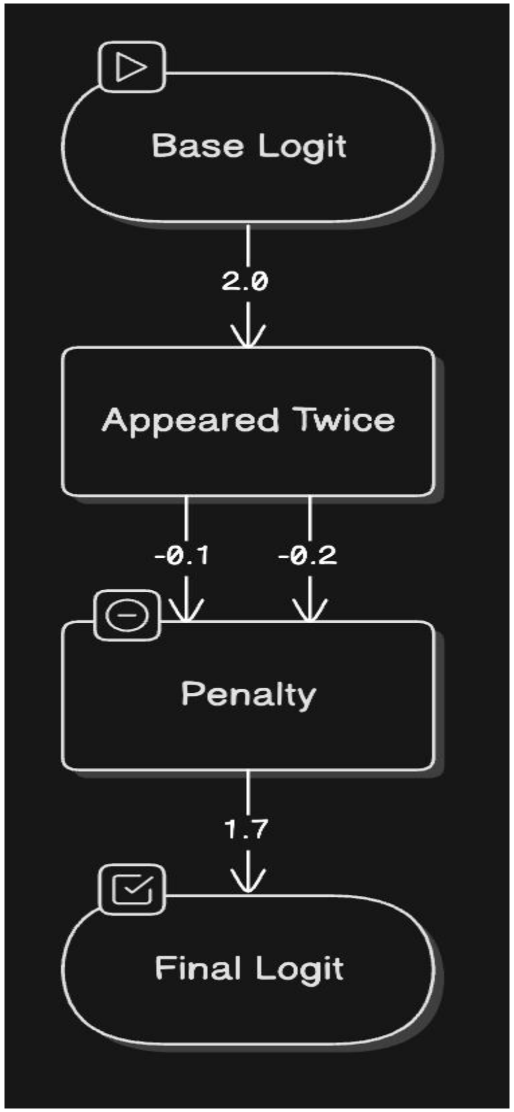
What sampling temperature to use, between 0 and 2. Higher values like 0.8 will make the output more random, while lower values like 0.2 will make it more focused and deterministic.

Softmax with Temperature


LLM logits for each token
Logit of i-th token
Total tokens in vocabulary
Temperature
Probability of i-th token
Token Probabilities (Low Temperature)
Token Probabilities (High Temperature)

$$P_i = \frac{e^{\frac{y_i}{T}}}{\sum_{j=1}^{V} e^{\frac{y_j}{T}}}$$

Repetitive Token Handling



1. **Presence Penalty**: A fixed penalty applied to any token that has appeared before, regardless of how often. This helps prevent the model from reusing the same words.
2. **Frequency Penalty**: A scaling penalty that increases based on how often a token has been used. The more a word appears, the less likely it is to be chosen again.