Deep Learning Frameworks

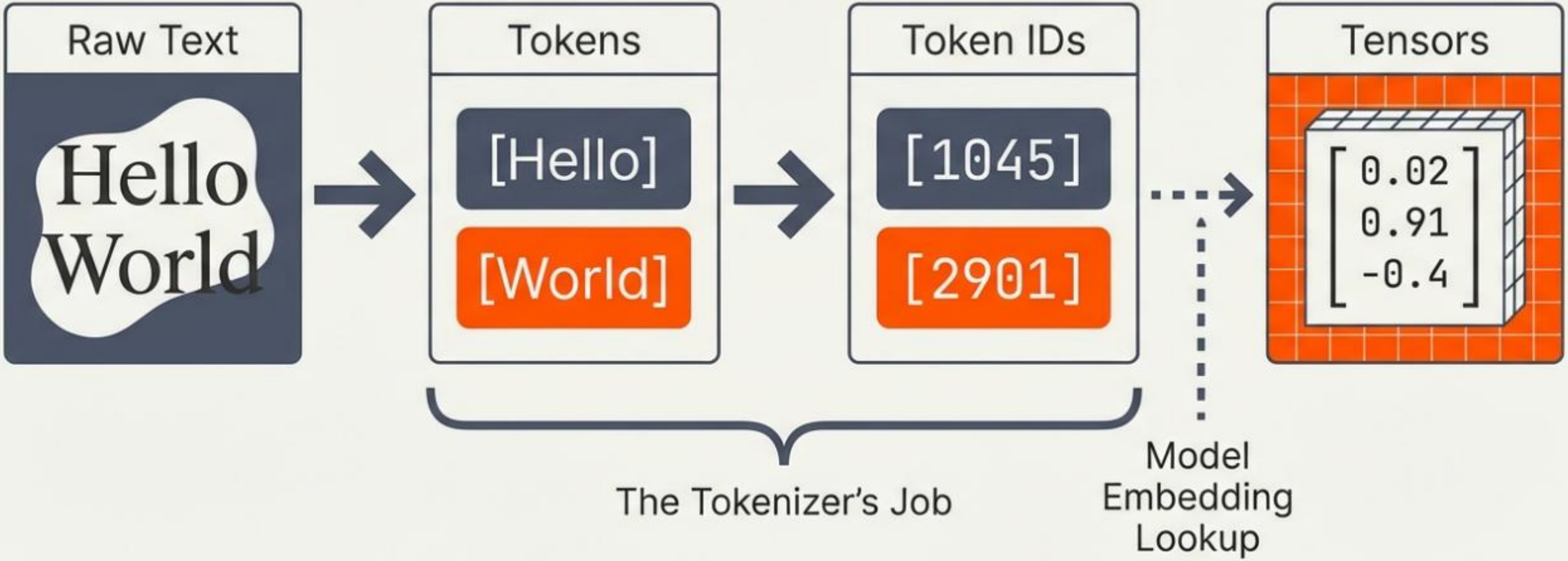
Tokenization, RNN, GRU, Attention

https://tinyurl.com/dlframeworks
https://github.com/sakharamg/DeepLearningFrameworks

Challenges with Text Data

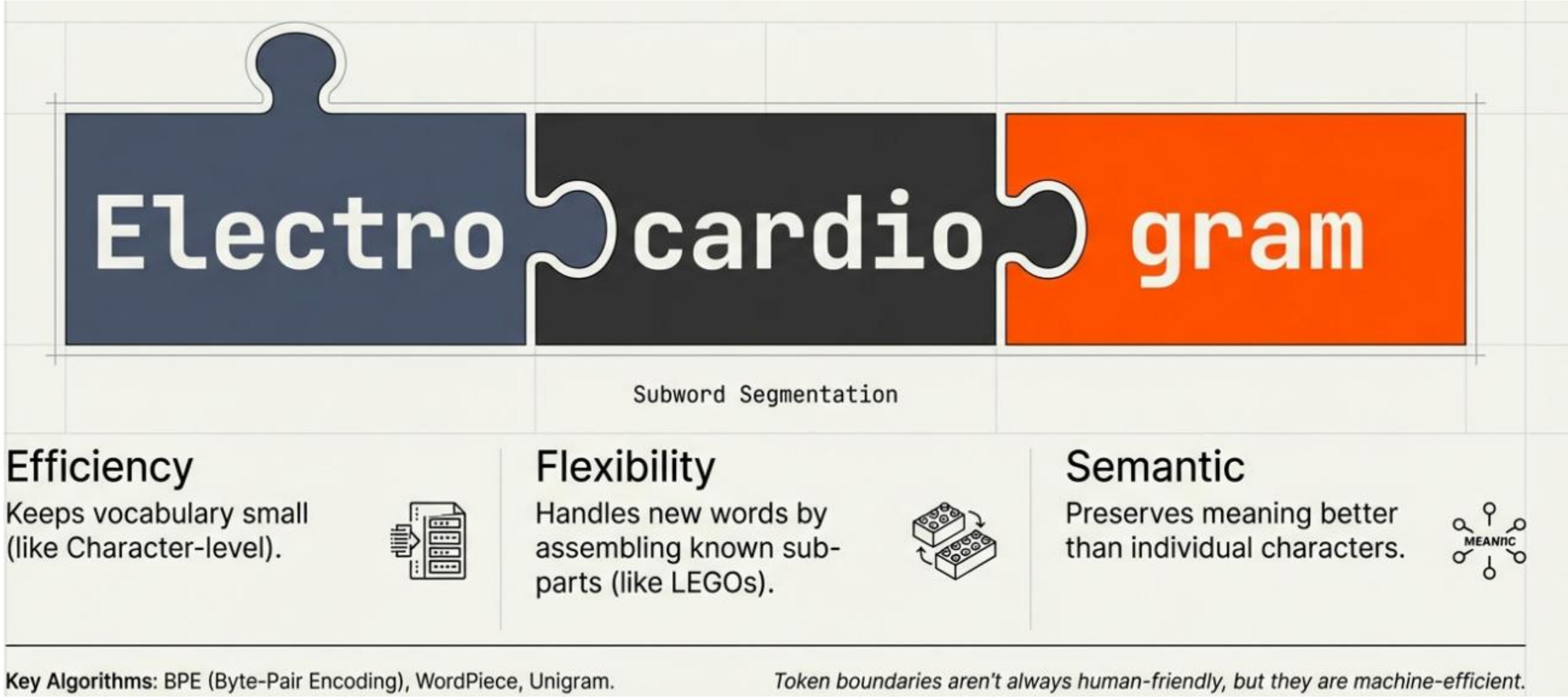
- Representing Text
- Varying Length

Representing Text

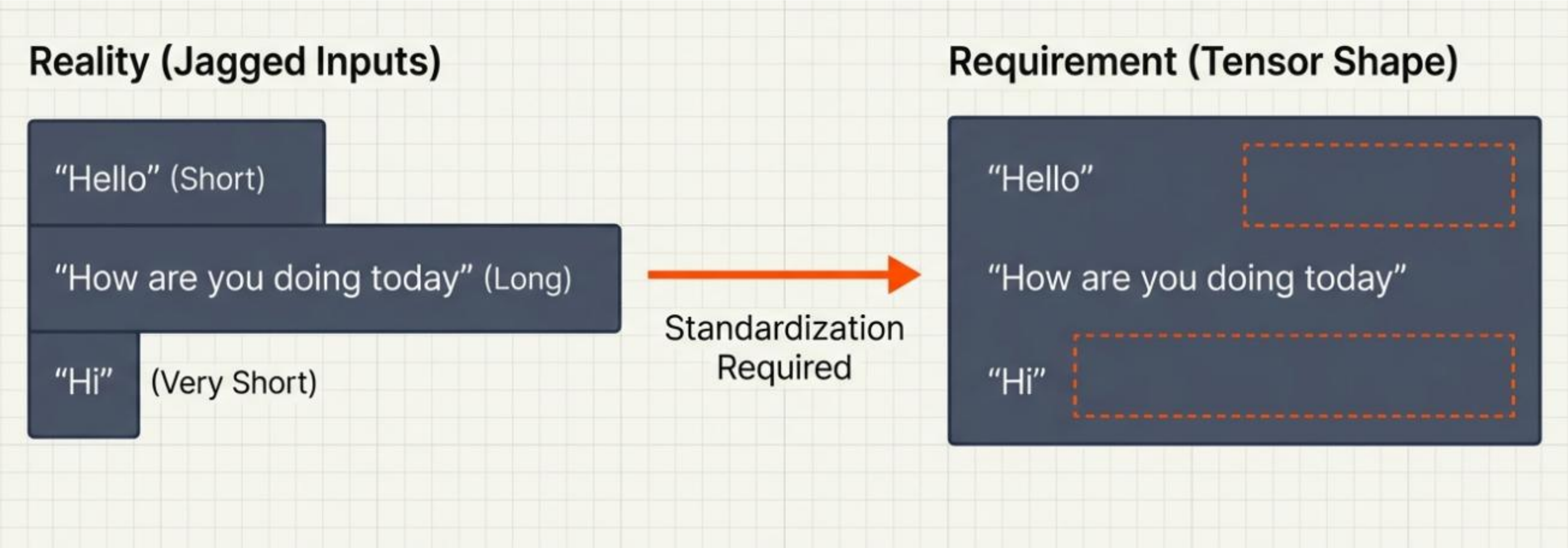


Tokenization: Splitting the text input

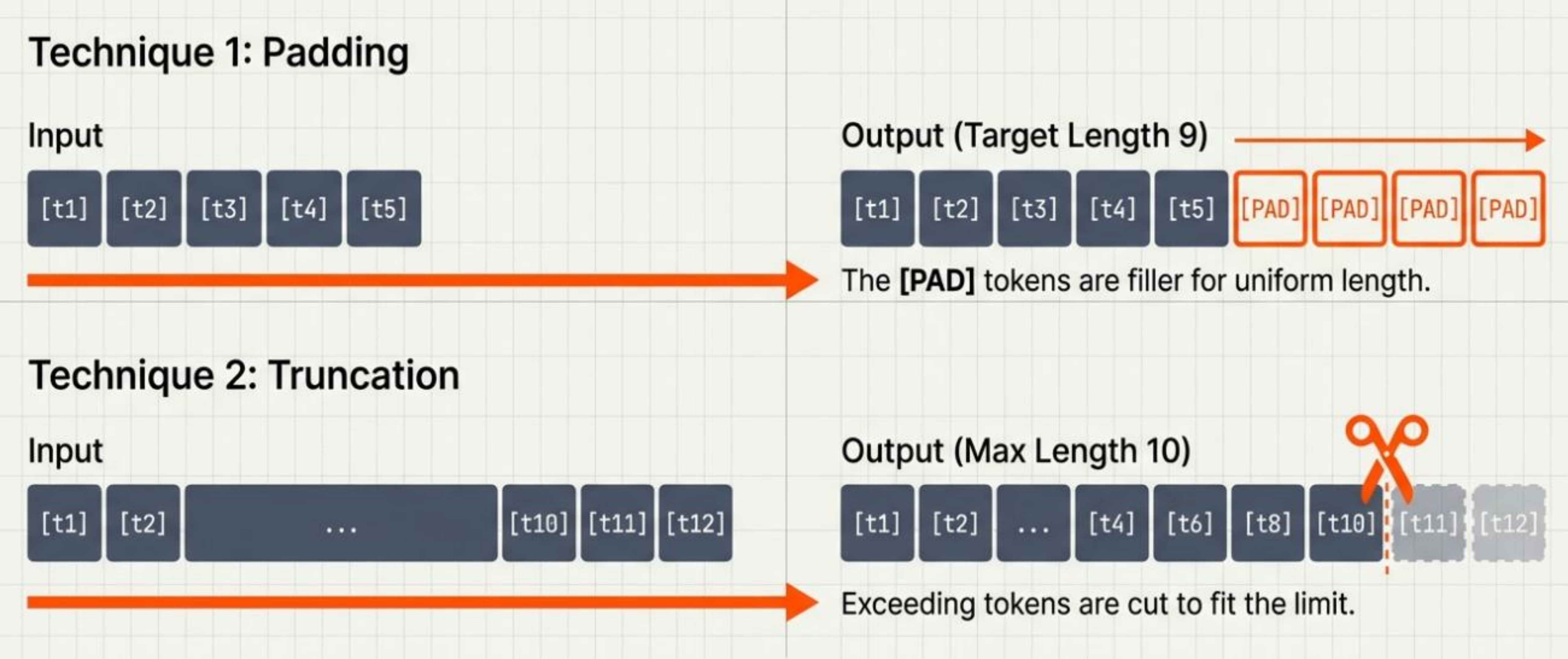
- Word level: linguistic boundaries, vocab explosion, missing words
- Character Level: Limited Vocab, No unknowns, Sequence length
- Subword Level



Handling varying lengths



Padding and Truncation



The Special Tokens Glossary

<PAD>

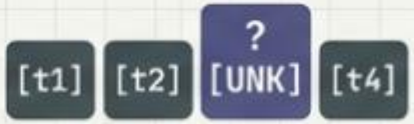
Padding

Fills empty space to maintain batch shape (rectangular constraint).

<UNK>

Unknown

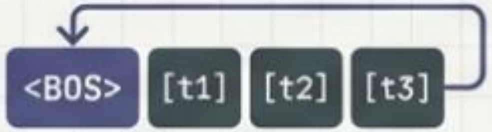
Represents out-of-vocabulary terms. Rare in modern subword models.



<BOS> / <CLS>

Begin-of-Sequence

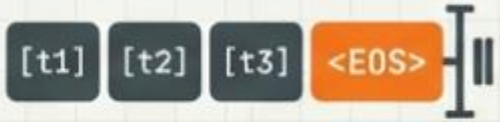
Marks the start of input. Often holds sentence-level context.



<EOS> / <SEP>

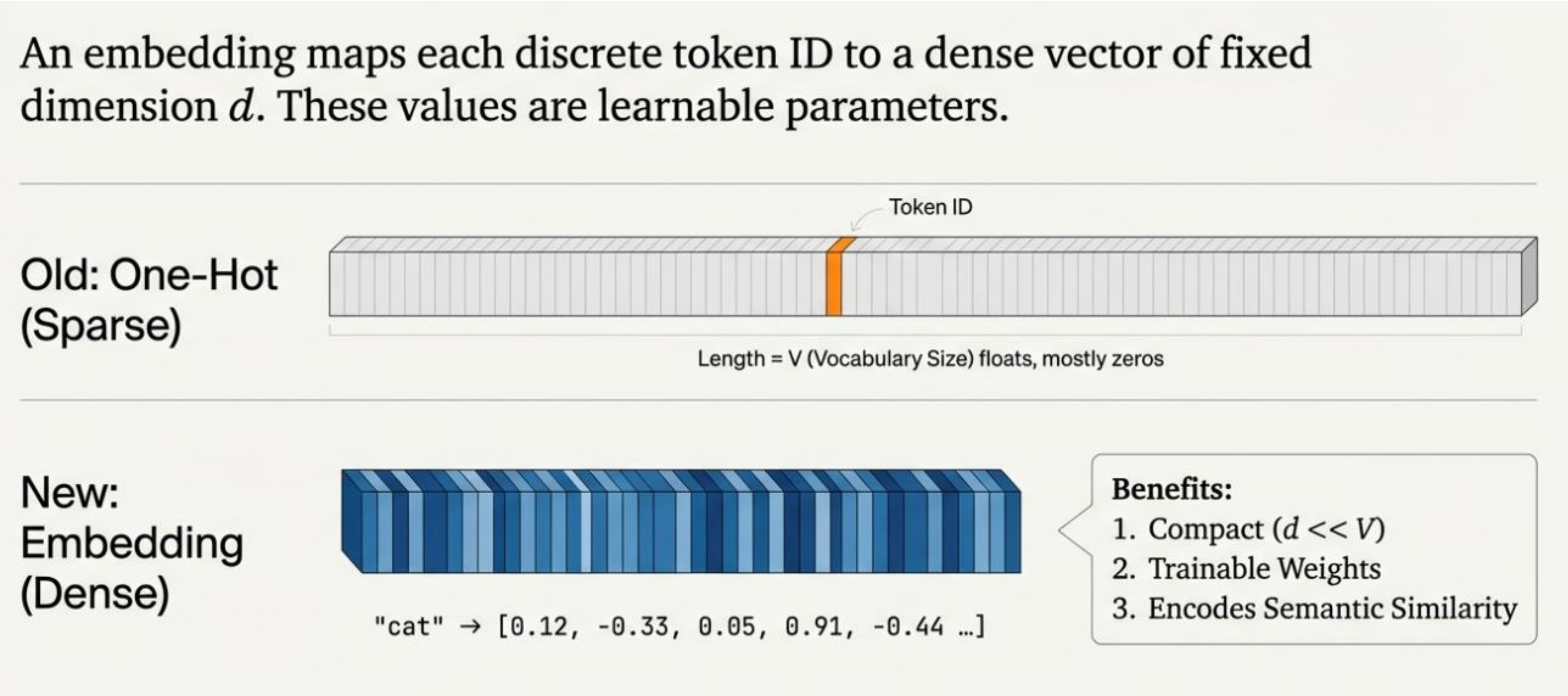
End-of-Sequence

Marks the conclusion of a segment or sentence.

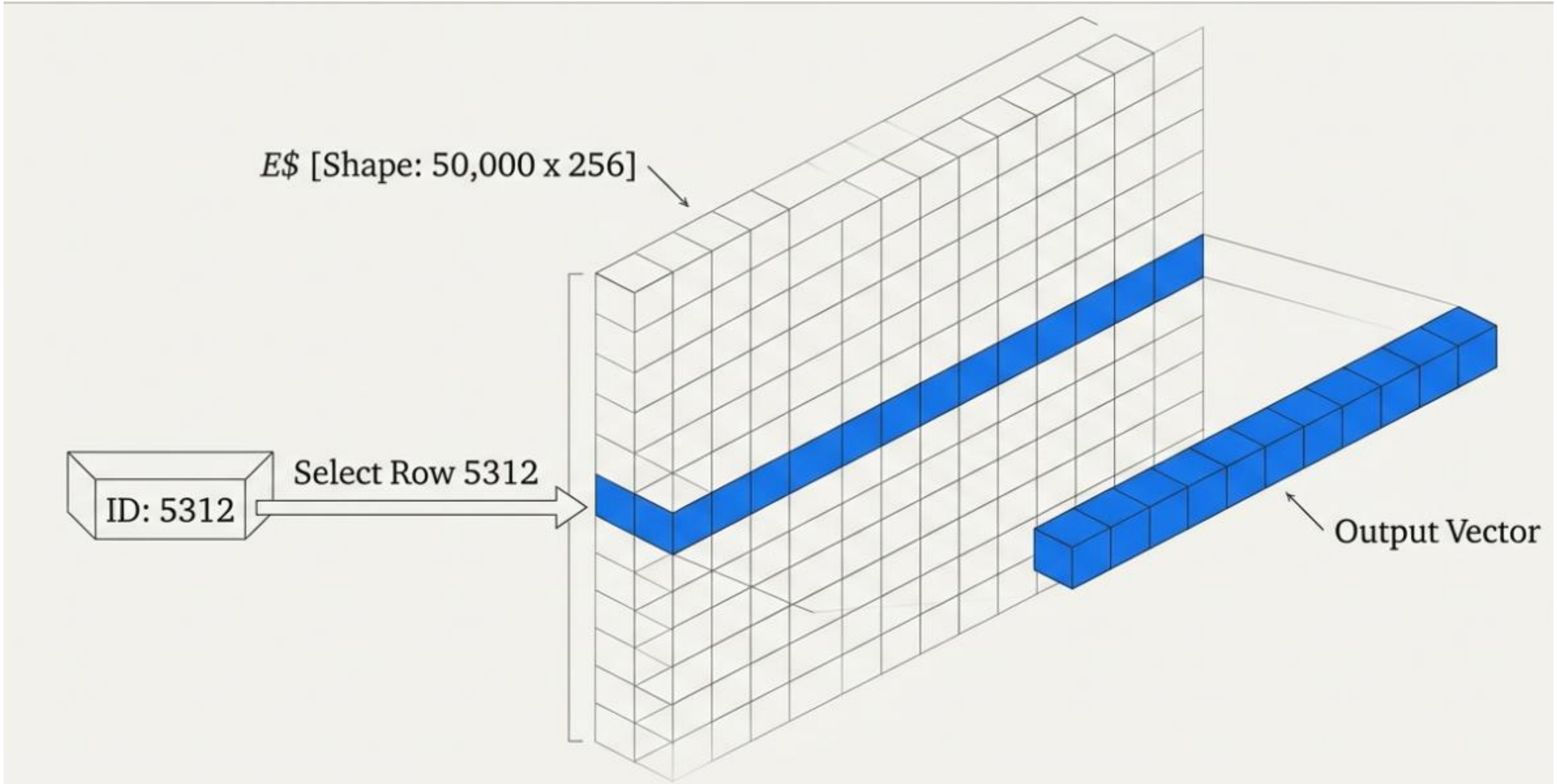


Ways to Represent

- Integers [512>8]
- One hot [1 0 0] -> orange [0 1 0]-> car



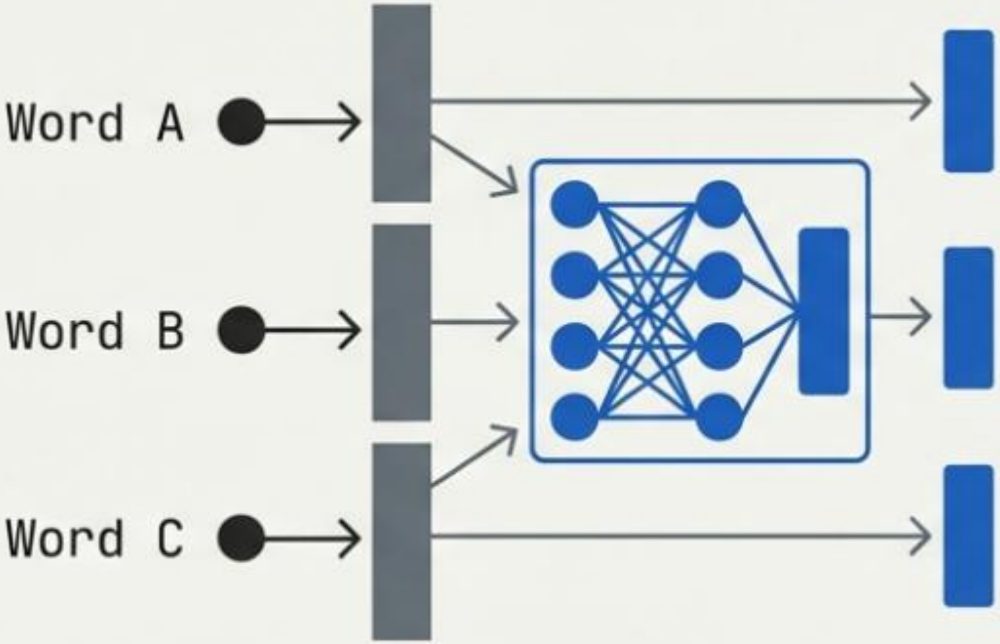
nn.Embedding



Sequence Matters

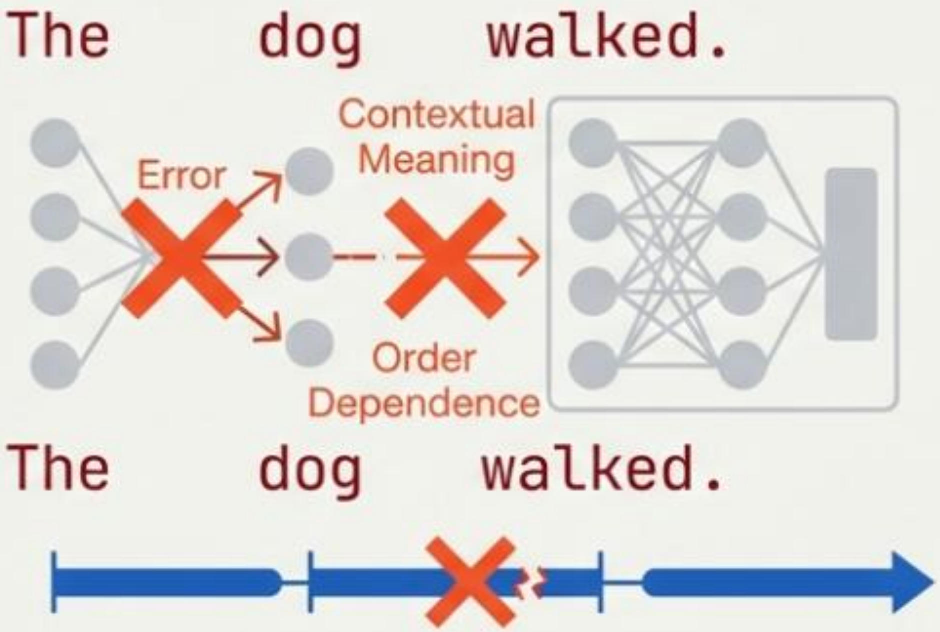
The Gap

Standard feedforward networks treat inputs as independent events. They have no concept of "before" or "after." A normal feedforward network cannot naturally "remember" what occurred in the previous step.



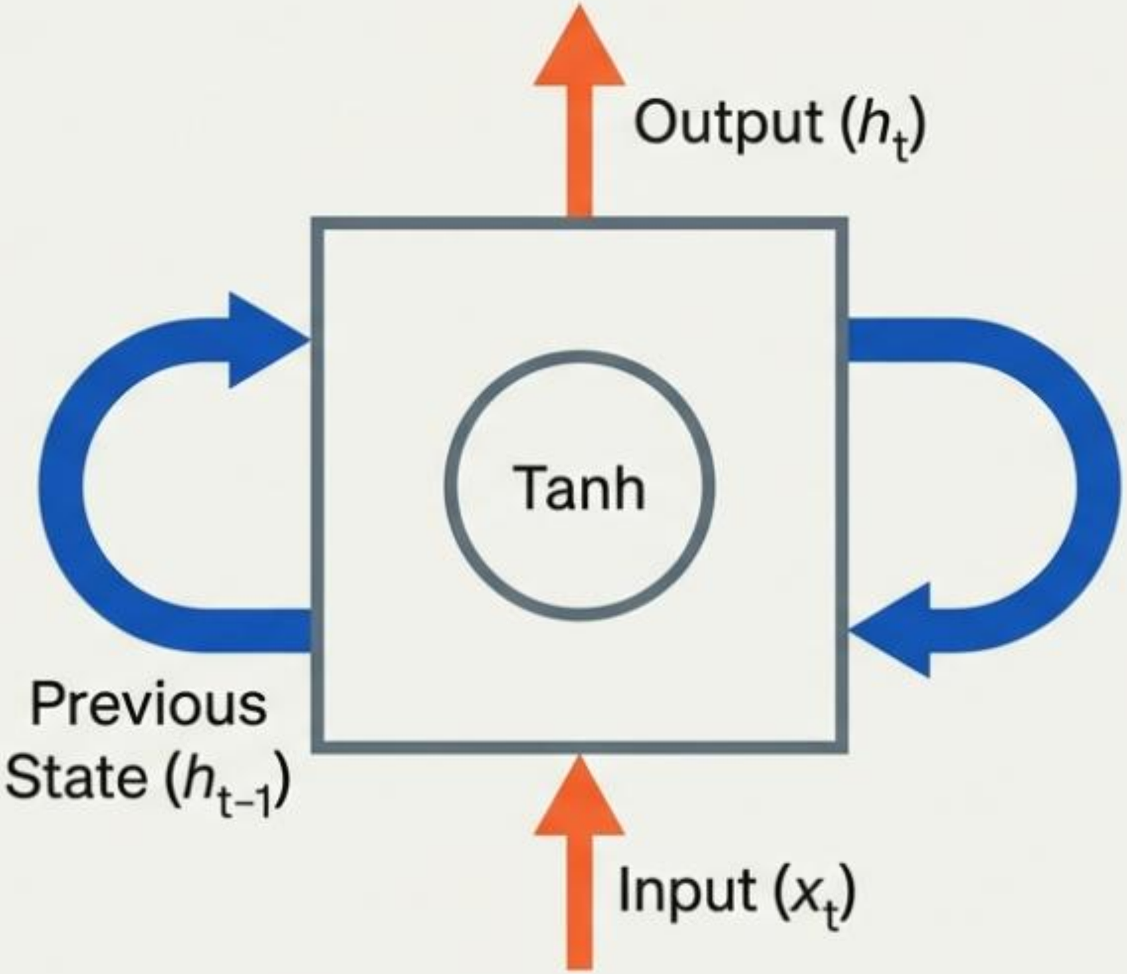
The Reality

Real-world data, such as text and time-series, is defined by continuity. Its key attributes are: Variable length, Order dependence, and Contextual meaning.



Recurrent Neural Network

Concept: The RNN processes tokens one step at a time while maintaining a 'hidden state' (h) that acts as memory.



Mechanism: At time step t, the RNN combines two signals:

1. The Current Input (x_t);
2. The Previous Hidden State (h_{t-1});

Result: A merger producing a new hidden state (h_t) that carries the past into the present.

State Update

Sensory Weights: The matrix processing the new input signal.

Memory Weights: The matrix processing the historical context (h_{t-1}).

$$h_t = \tanh(W_{xh}x_t + W_{hh}h_{t-1} + b_h)$$

Activation: The non-linear function squashing values between -1 and 1.

Bias: The learnable offset.

The new state is a weighted sum of immediate sensory data and historical context, fused mathematically by the tanh activation.

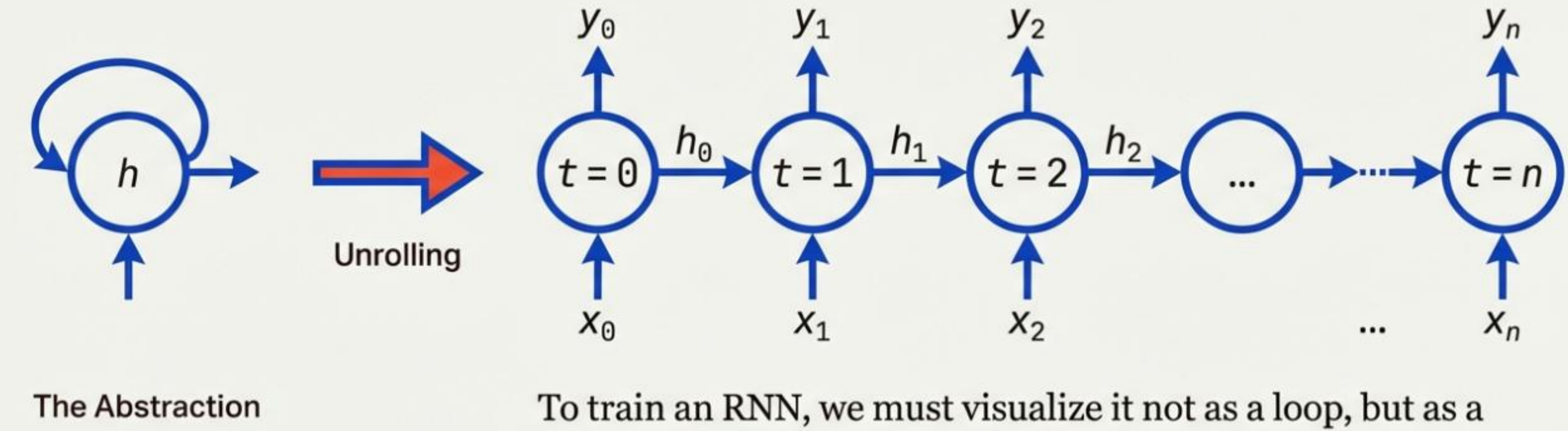
Output

$$y_t = W_{hy}h_t + b_y$$

The hidden state h_t is internal memory. To get a prediction (like the next word), we project h_t through an output weight matrix (W_{hy}).

Note: This output step is optional at intermediate steps, depending on the task topology.

Unroll

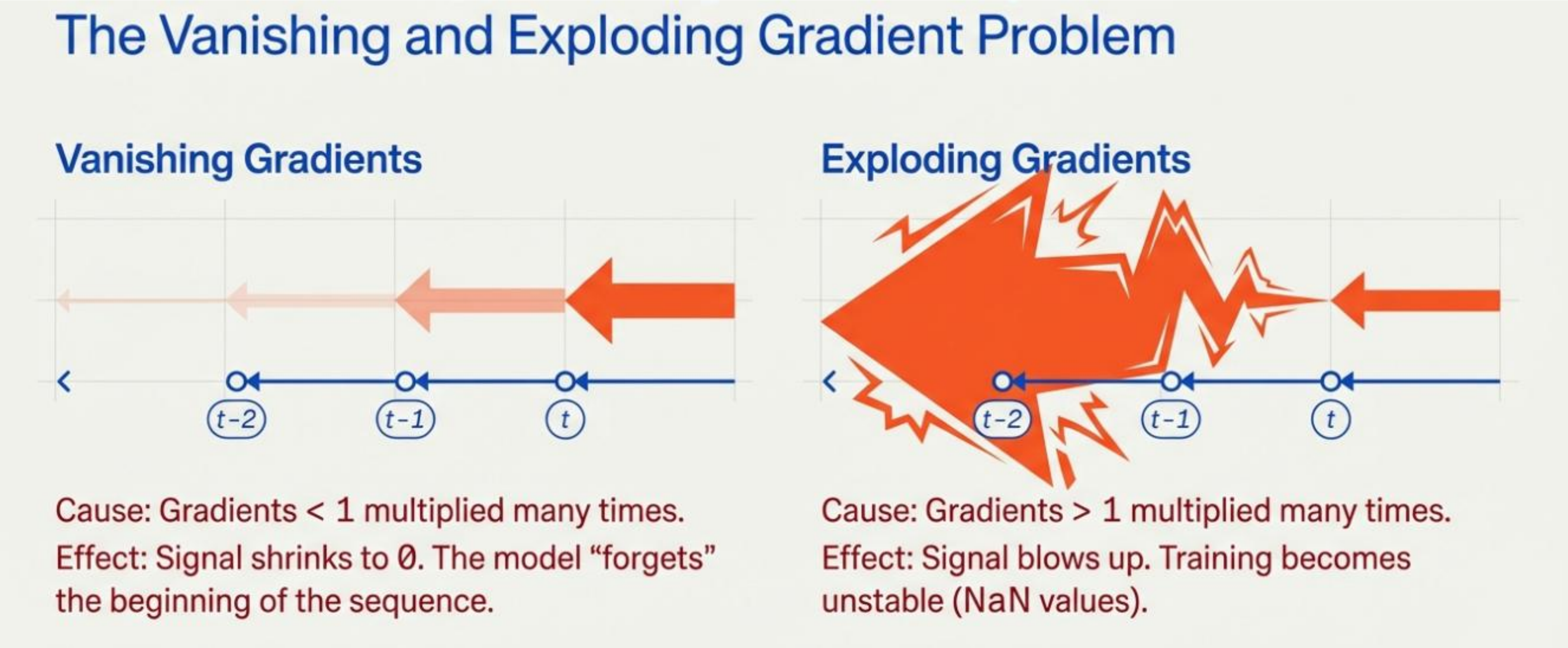


To train an RNN, we must visualize it not as a loop, but as a deep neural network where each layer corresponds to a time step. A sequence of 100 words is effectively a 100-layer deep network sharing the same weights.

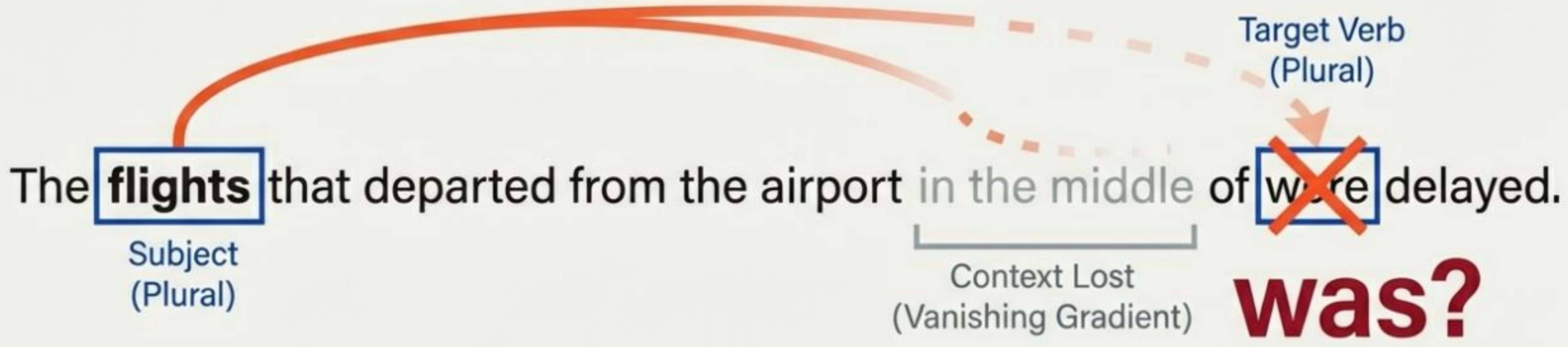
Encoder

Encoder - Decoder

Instability in Long Term Dependencies

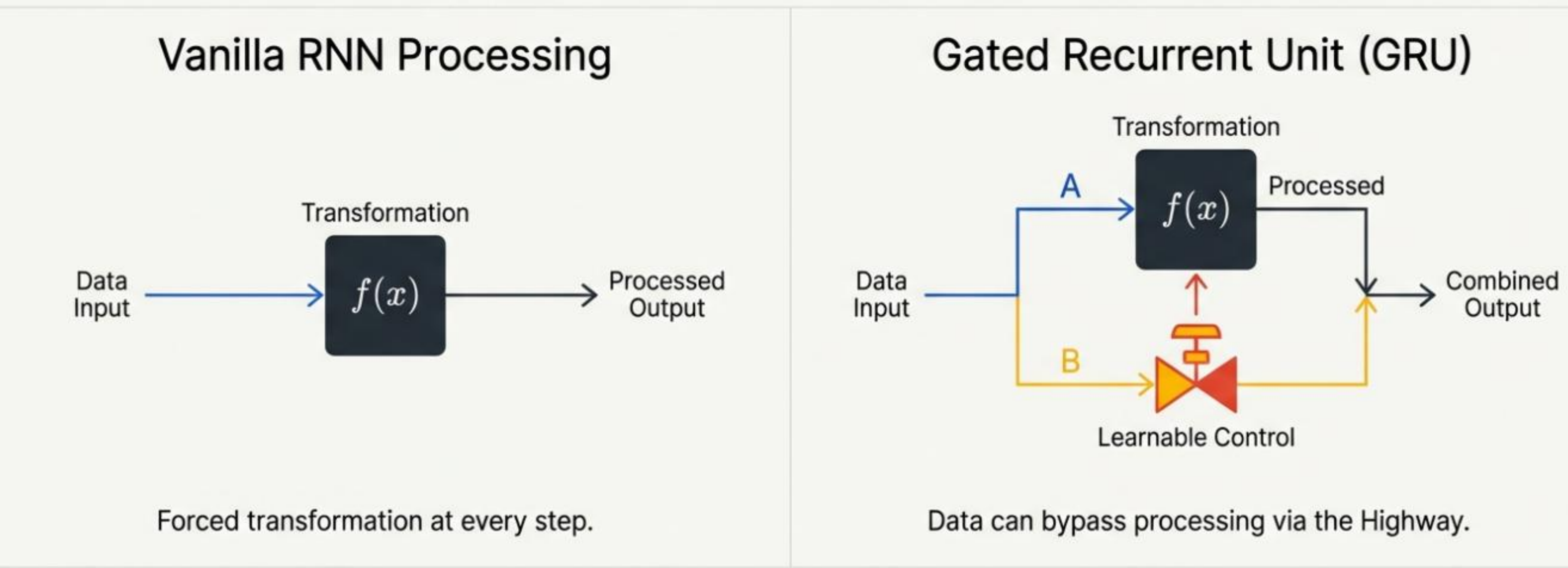


Synptoms of Memory Loss



The Symptom: The model learns short-term dependencies well but fails at long contexts. Here, the RNN forgets the plural subject “flights” by the time it reaches the verb, potentially predicting the singular “was”.

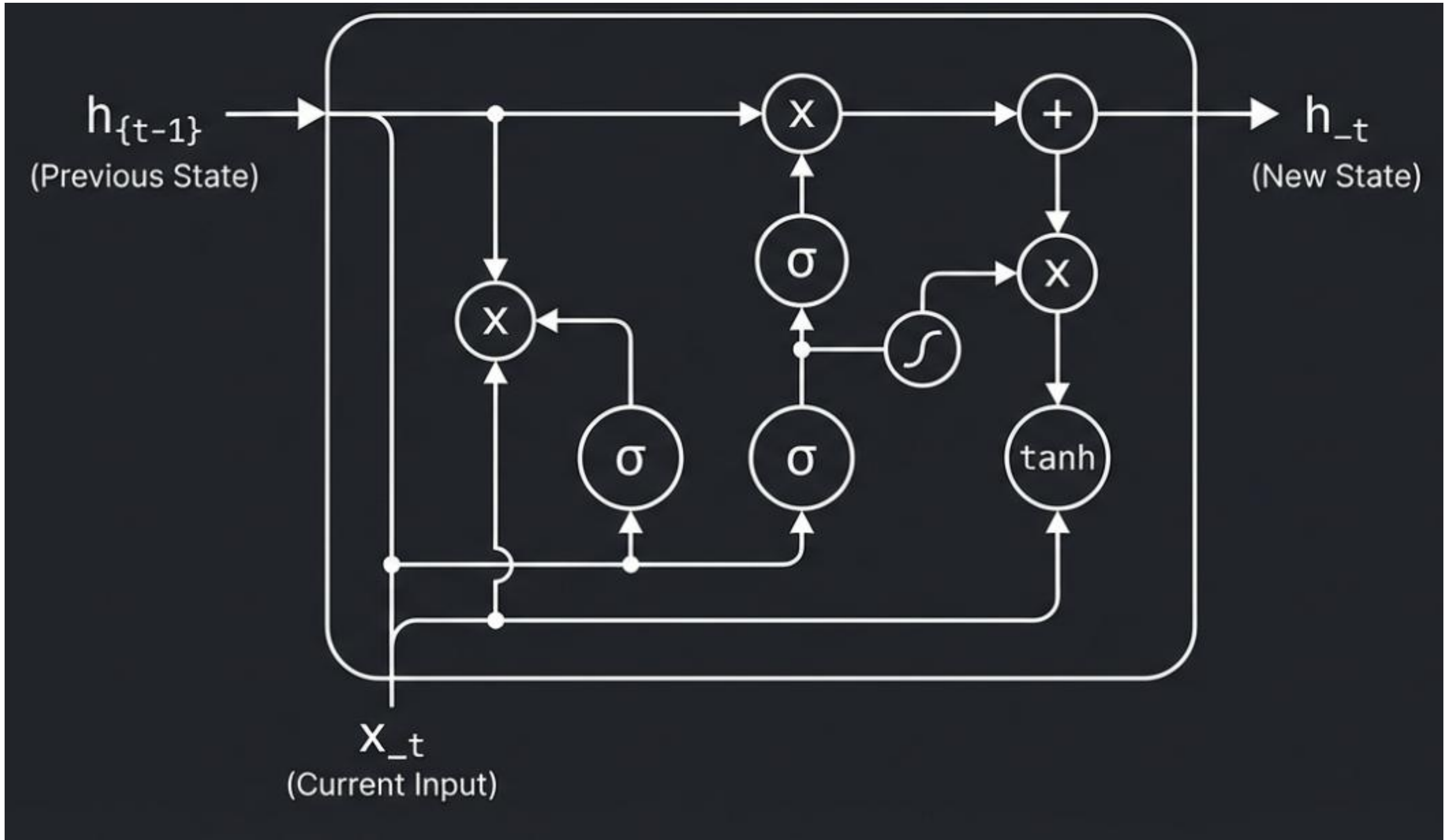
Handling Long Term Dependencies



Core Concept: Instead of forcing the model to rewrite its entire memory at every step, we add learnable **Gates** (z_t, r_t). These differentiable knobs learn when to **Keep** existing memory, **Update** with new input, or **Ignore** irrelevant history. The GRU is a streamlined evolution of the LSTM, using only 2 gates instead of 3.

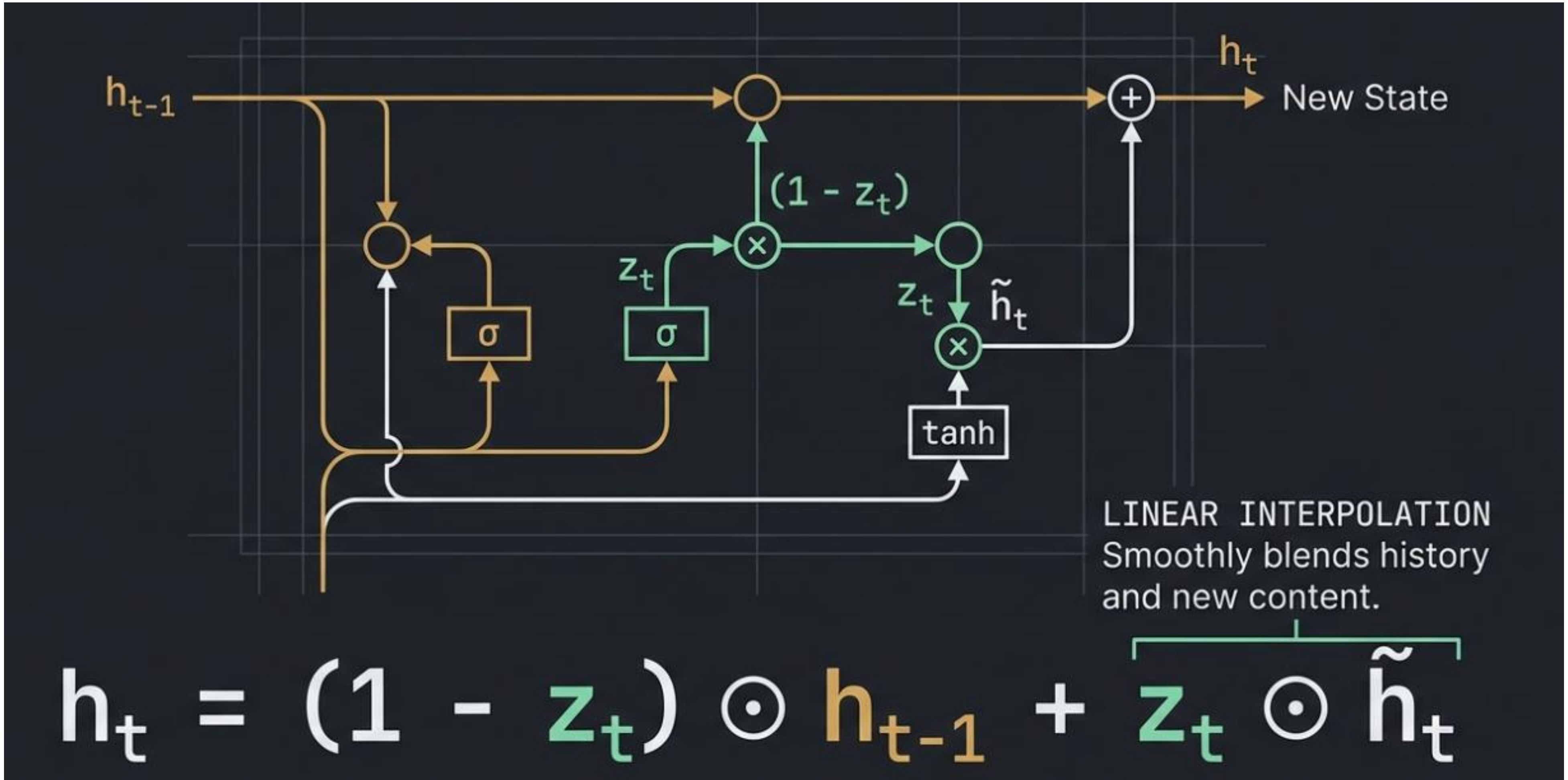
GRU

The delivery was two days late and the box was torn. Customer support was slow to respond. **Anyway, the Bluetooth earbuds paired instantly with my phone. The sound is clear, bass is decent, and the battery easily lasts 6–7 hours.** I used them during a workout and the fit stayed secure.



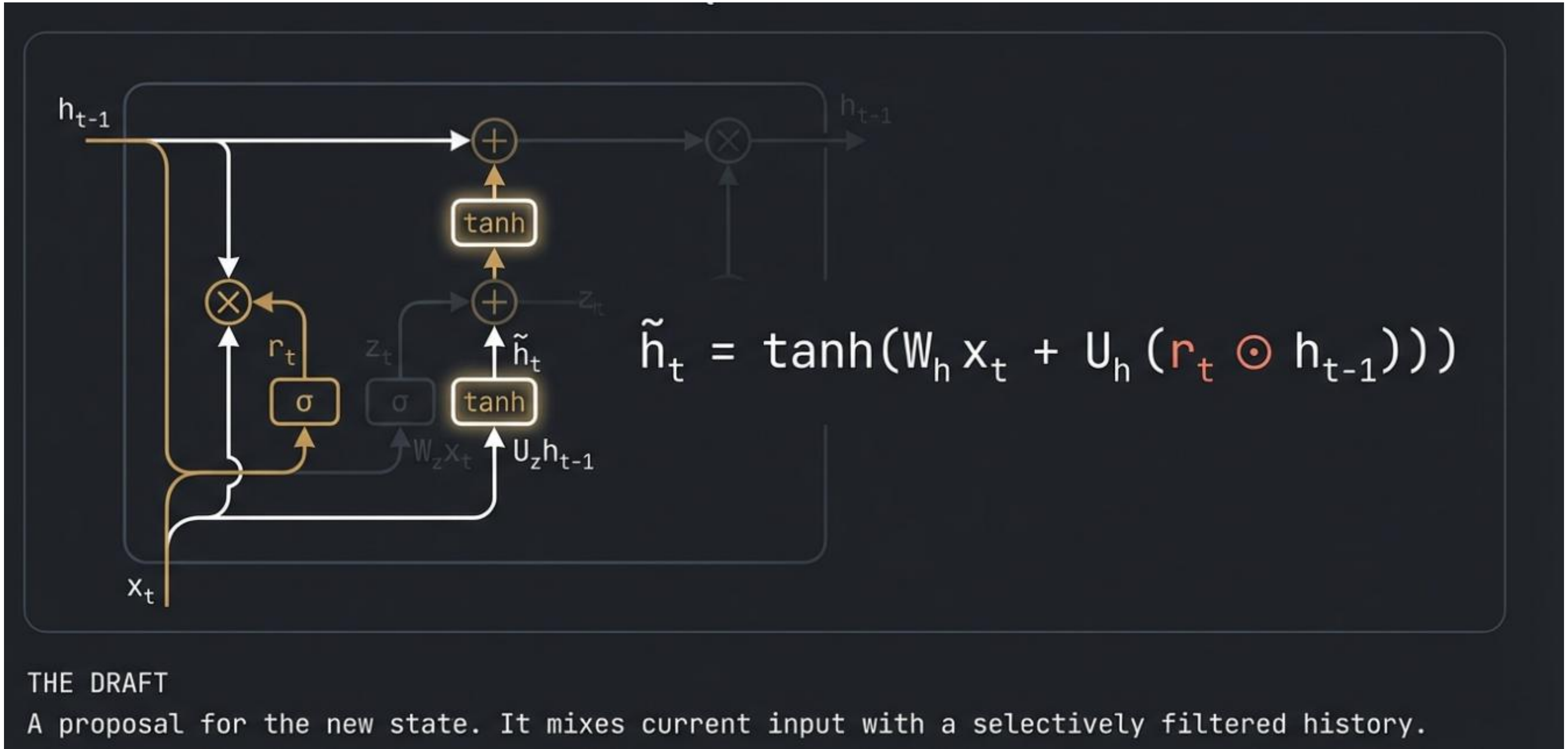
Final State: How much should I keep vs replace?

The delivery was two days late and the box was torn. Customer support was slow to respond. **Anyway, the Bluetooth earbuds paired instantly with my phone. The sound is clear, bass is decent, and the battery easily lasts 6–7 hours.** I used them during a workout and the fit stayed secure.

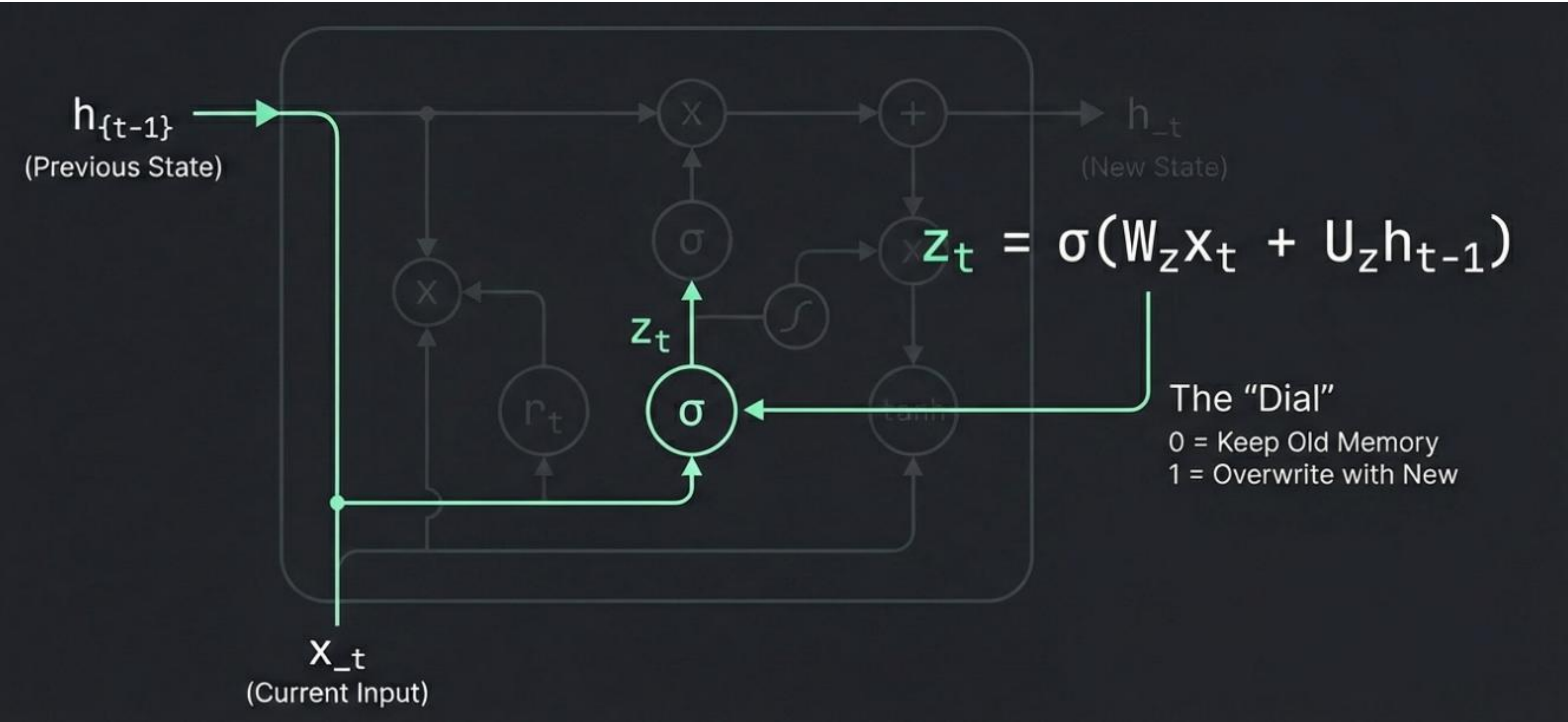


Candidate State

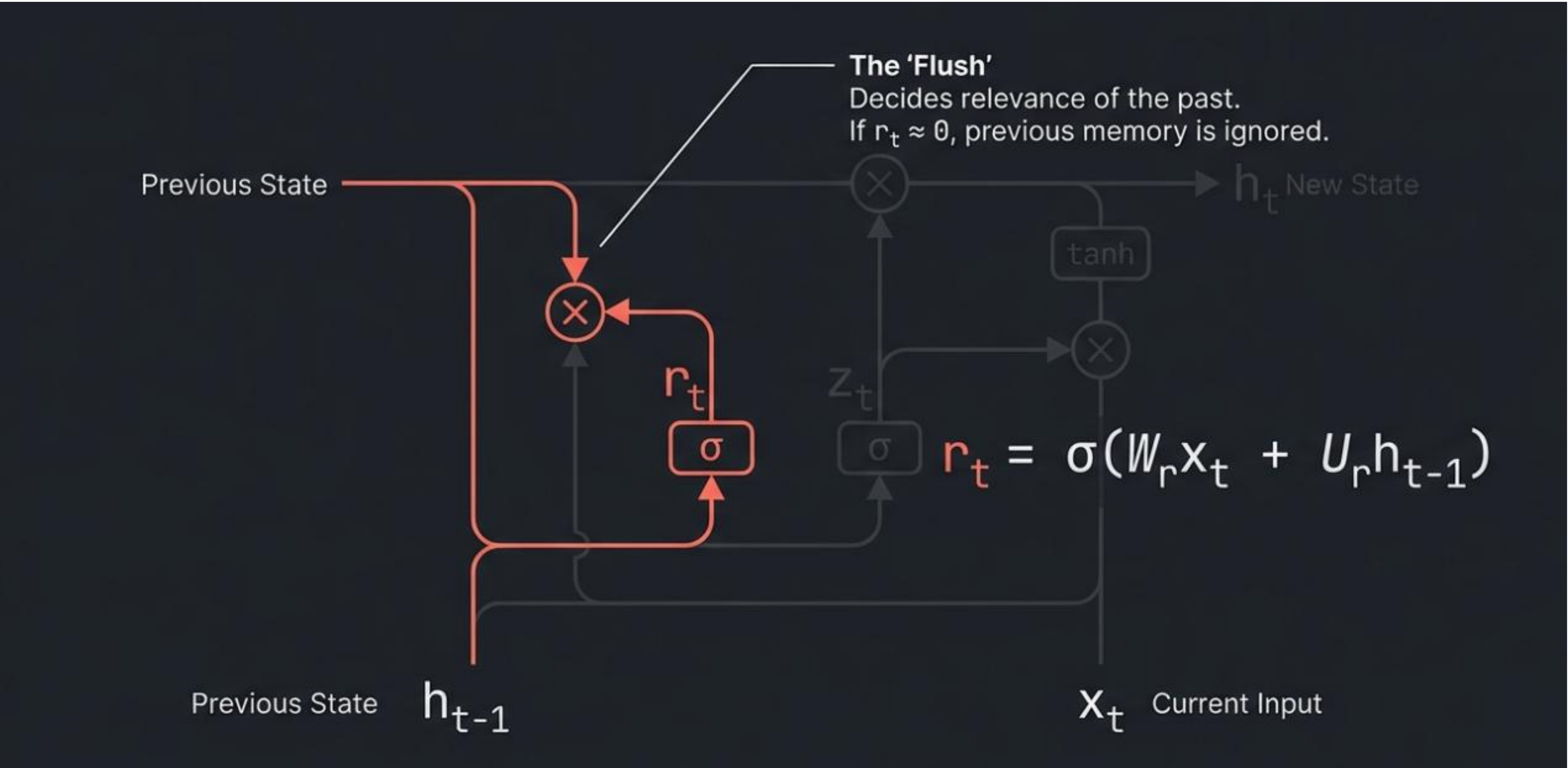
The delivery was two days late and the box was torn. Customer support was slow to respond. **Anyway, the Bluetooth earbuds paired instantly with my phone. The sound is clear, bass is decent, and the battery easily lasts 6–7 hours.** I used them during a workout and the fit stayed secure.



Update Gate

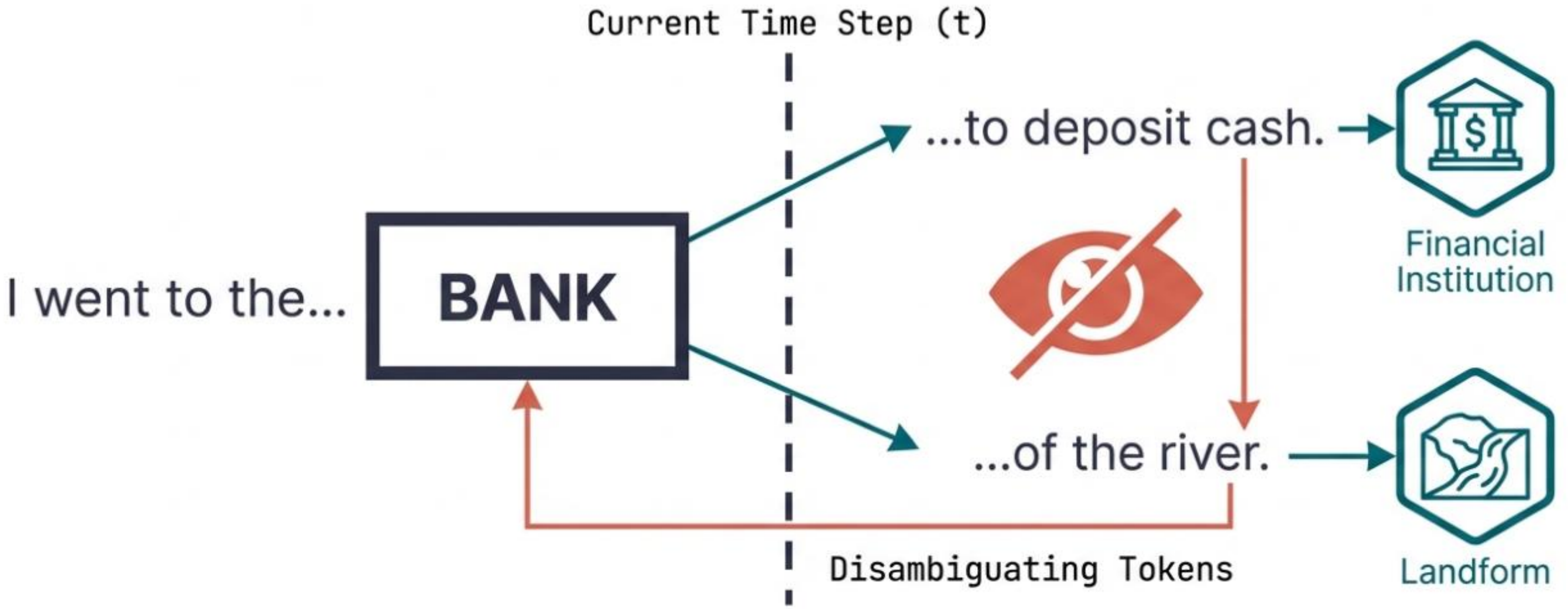

h_{t-1}
(Previous State)
h_t
(New State)
$z_t = \sigma(W_z x_t + U_z h_{t-1})$
z_t
σ
The "Dial"
0 = Keep Old Memory
1 = Overwrite with New
x_t
(Current Input)

Reset Gate

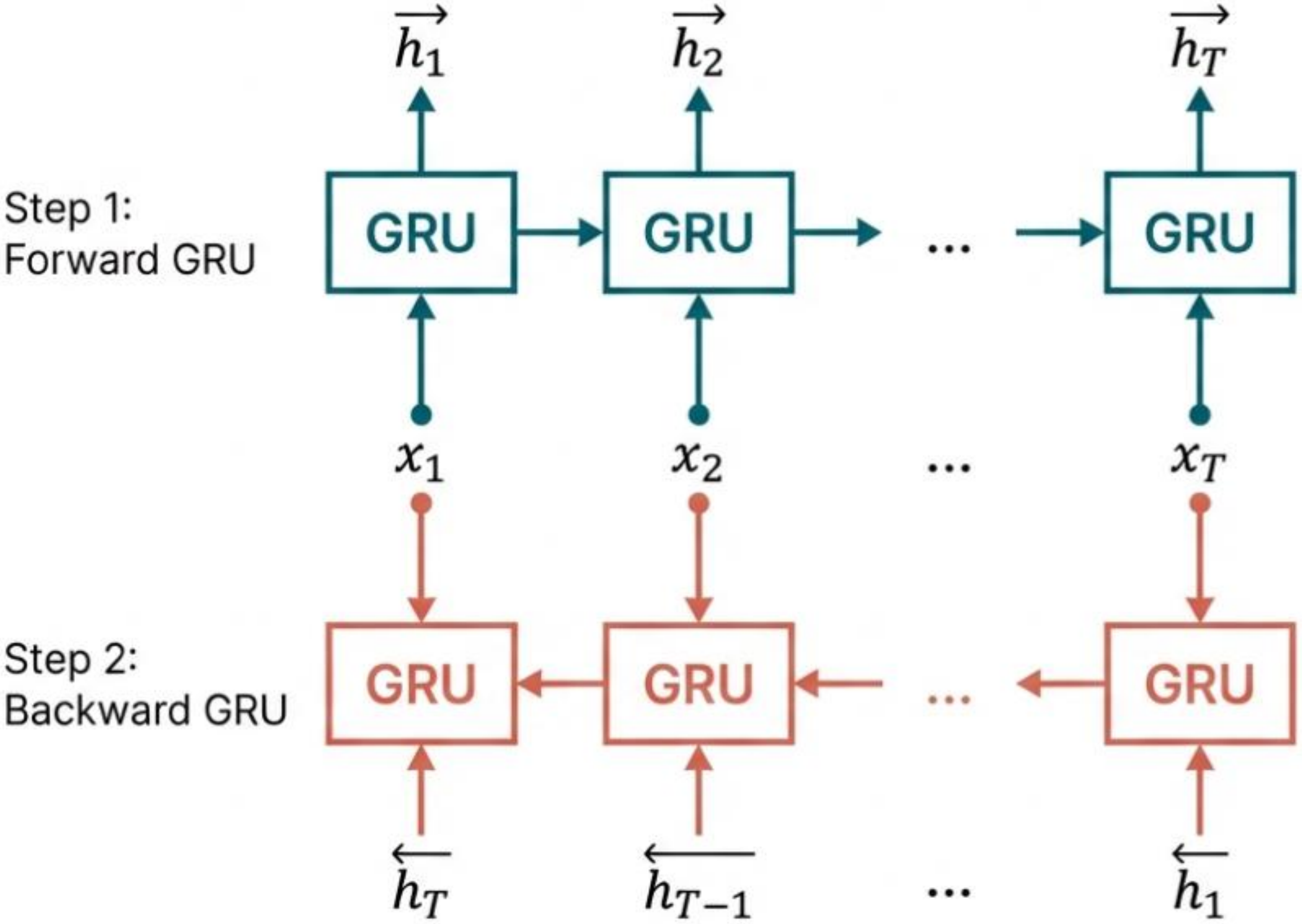

The 'Flush'
Decides relevance of the past.
If $r_t \approx 0$, previous memory is ignored.
Previous State
h_t New State
tanh
r_t
z_t
σ
σ
$r_t = \sigma(W_r x_t + U_r h_{t-1})$
Previous State h_{t-1}
x_t Current Input

Future gives the full picture

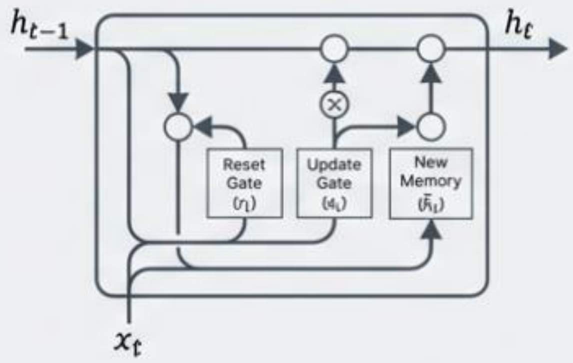
In sequence processing, the meaning of a specific token frequently depends on both its left (past) and right (future) context. Without the full picture, interpretation fails.



Bi Directional GRU

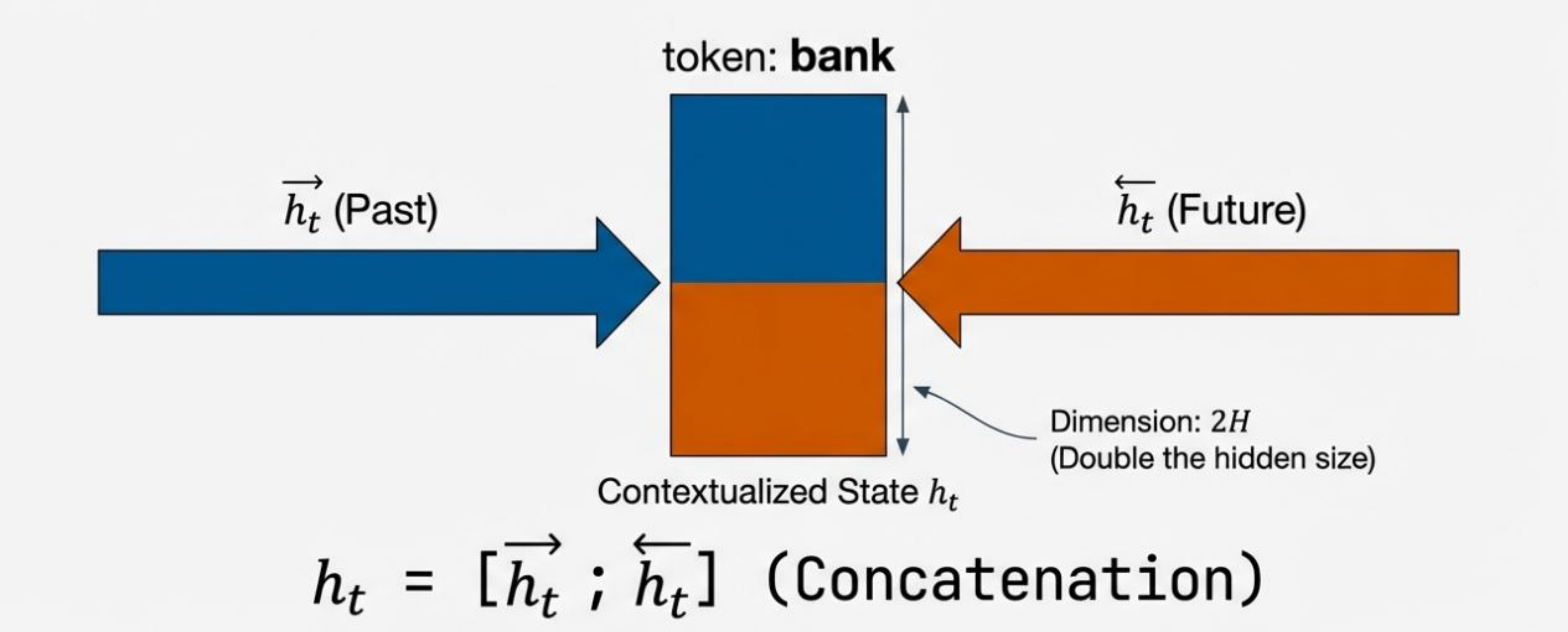


Inside the GRU



Gates control information flow, deciding what to keep from the past and what to overwrite.

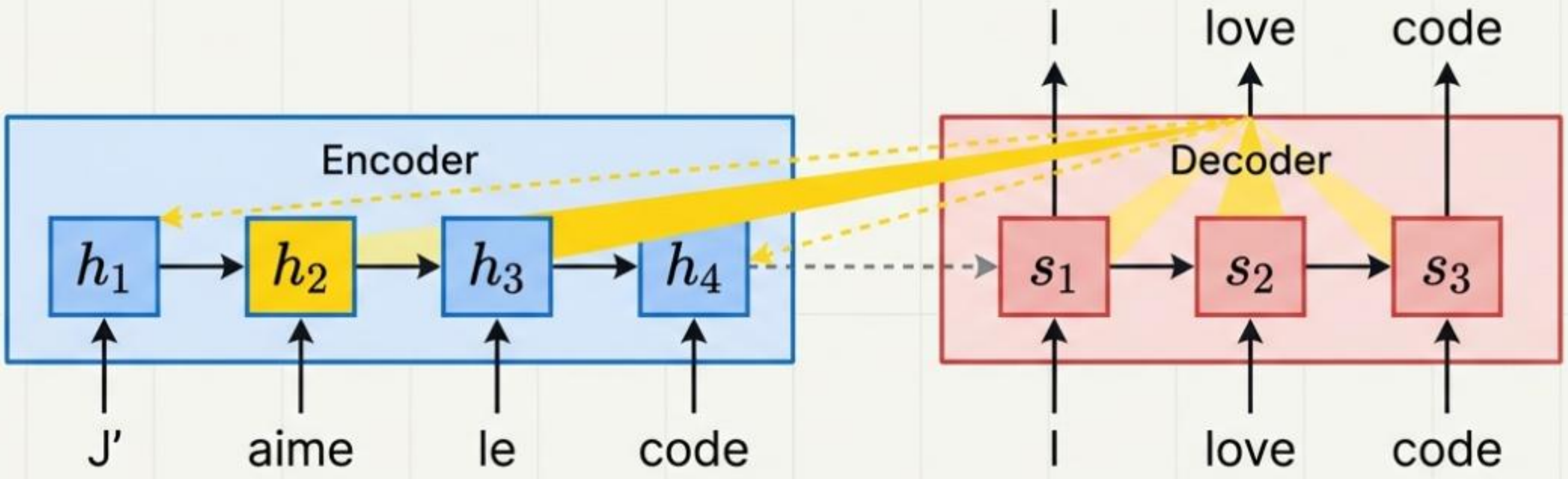
Merging States from Both Direction



Lab

https://tinyurl.com/dlframeworks
https://github.com/sakharamg/DeepLearningFrameworks

Attention



The Solution:: Instead of relying on a static summary, we allow the decoder to **'look back'** at the entire history of encoder states.

Dynamic Context: At every step of decoding, the model creates a unique context vector relevant *only* to that specific moment.

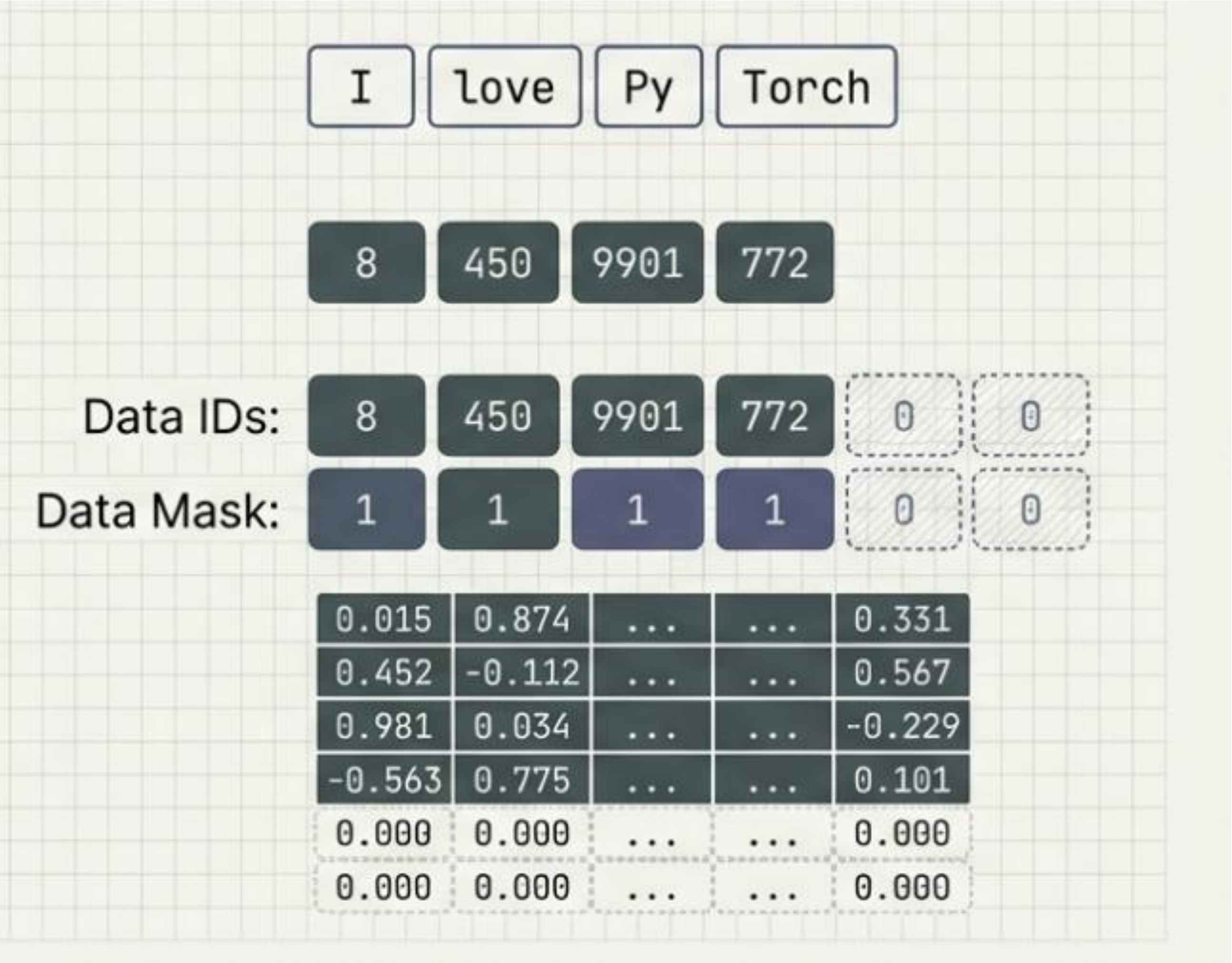
Approach

Lab

https://tinyurl.com/dlframeworks
https://github.com/sakharamg/DeepLearningFrameworks

Thank You

Appendix



I
love
Py
Torch
8
450
9901
772
Data IDs:
8
450
9901
772
0
0
Data Mask:
1
1
1
1
0
0
0.015 0.874 ... ... 0.331
0.452 -0.112 ... ... 0.567
0.981 0.034 ... ... -0.229
-0.563 0.775 ... ... 0.101
0.000 0.000 ... ... 0.000
0.000 0.000 ... ... 0.000