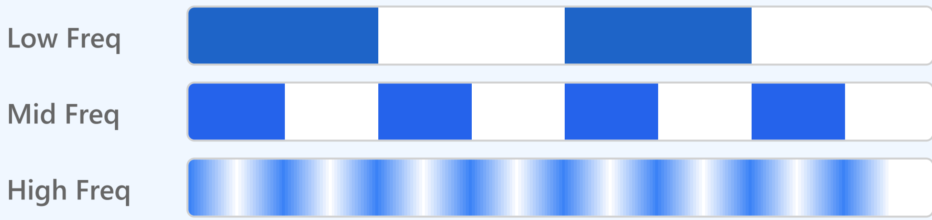
Positional Encoding Visualization

🌊 Frequency Patterns Across Dimensions



🌍 Low Frequencies

Capture **global position**
Overall sequence location

🎯 High Frequencies

Capture **local position**
Fine-grained distinctions

🔑 Unique Patterns

Each position has **unique encoding pattern**

📈 Smooth Gradient

Allows model to learn **relative positions**

🎵 Multiple Frequencies

Different frequencies for different dimensions

⚖️ Magnitude Balance

Magnitude **comparable** to embedding values

Element-wise addition to token embeddings | Preserves both semantic and positional information

📊 Calculation Example (d_model=4, max_len=8)

Formula

$$PE_{(pos,2i)} = \sin(pos / 10000^{2i/d\_model})$$

$$PE_{(pos,2i+1)} = \cos(pos / 10000^{2i/d\_model})$$

🔤 Token Embeddings

📍 Positional Encoding

✨ Final Input

X (8×4)

0.5	-0.3	0.8	-0.2
-0.1	0.7	0.2	0.4
0.9	-0.5	-0.3	0.6
0.3	0.4	-0.7	0.1
-0.6	0.2	0.5	-0.4
0.7	-0.8	0.1	0.3
-0.2	0.6	-0.4	0.8
0.4	-0.1	0.9	-0.5

+

PE (8×4)

0.00	1.00	0.00	1.00
0.84	0.54	0.01	1.00
0.91	-0.42	0.02	1.00
0.14	-0.99	0.03	1.00
-0.76	-0.66	0.04	1.00
-0.96	0.28	0.05	1.00
-0.28	0.96	0.06	1.00
0.66	0.75	0.07	0.99

=

X + PE (8×4)

0.50	0.70	0.80	0.80
0.74	1.24	0.21	1.40
1.81	-0.92	-0.28	1.60
0.44	-0.59	-0.67	1.10
-1.36	-0.46	0.54	0.60
-0.26	-0.52	0.15	1.30
-0.48	1.56	-0.34	1.80
1.06	0.65	0.97	0.49

💡 **Key Insight:** Each position gets a unique sinusoidal pattern. Lower dimensions (0,1) use **high frequency** for fine-grained position, while higher dimensions (2,3) use **low frequency** for global position.