

```
def perceiver_resampler(
    x_f, # The [T, S, d] visual features (T=time, S=space)
    time_embeddings, # The [T, 1, d] time pos embeddings.
    x, # R learned latents of shape [R, d]
    num_layers, # Number of layers
):
  """The Perceiver Resampler model."""
  # Add the time position embeddings and flatten.
  x_f = x_f + time_embeddings
  x_f = flatten(x_f) \# [T, S, d] \rightarrow [T * S, d]
  # Apply the Perceiver Resampler layers.
  for i in range(num_layers):
    # Attention.
    x = x + attention_i(q=x, kv=concat([x_f, x]))
    # Feed forward.
    x = x + ffw_i(x)
  return x
```