**Algorithm: Proxy-Tuning of Language Model**

/\* Applies proxy-tuning to adjust logits of a target model in an NLP task \*/

**Input**: input\_ids, model\_base, model\_tuned, model\_target

**Output**: generated\_text, a sequence of tokens

**Hyperparameters**: max\_length (maximum generation length), n (number of tokens to generate)

**Parameters**: Θ includes all parameters for the base, tuned, and target models.

1. Initialize generated\_tokens as an empty list

2. Encode input\_text into input\_ids using tokenizer Θ\_tokenizer

// Perform token-wise proxy-tuning and text generation

3. For t in [1, ..., n]:

a. Obtain base, tuned, and target model logits:

i. logits\_base ← model\_base(input\_ids).logits with parameters Θ\_base

ii. logits\_tuned ← model\_tuned(input\_ids).logits with parameters Θ\_tuned

iii. logits\_target ← model\_target(input\_ids).logits with parameters Θ\_target

b. Proxy-tuning adjustment:

i. Δlogits ← logits\_tuned - logits\_base

ii. logits' ← logits\_target + Δlogits

c. Normalize the logits for next token prediction:

i. predictions ← softmax(logits', axis=-1)

d. Select the next token:

i. next\_token\_id ← argmax(predictions)

ii. Append next\_token\_id to generated\_tokens

e. Update input\_ids with next\_token\_id for the next iteration

4. Decode the sequence of generated\_tokens into text using Θ\_tokenizer

5. **Return** generated\_text