Supplementary Material

1 Algorithm for Data Reconstruction

Algorithm 1 Algorithm for constructing D_r^t from M^{t-1}.

```
Input: Model from the previous step, $M^{t-1}$, set of old classes up to $t-1$, $V = \{v_i\}_{i=1}^{C^{t-1}}$.
Output: The reconstructed data $D_r^t$.
$D_r^t = \emptyset$
for v in V do
  for $i$ in $1 \cdots N$ do
    Uniformly sample $n_e \in [1, n_e^{max}]$.
    Uniformly sample $n_s \in [n_e, n_s^{max}]$.
    Uniformly sample $k \in [1, n_s - n_e + 1]$.
    Construct a target label sequence $Y$ of length $n_s$,
    with a length $n_e$ entity of class $v$ starting from position $k$.
    Randomly initialize an embedding sequence $E$ of length $n_s$.
    while not converge do
      Update $E$ with eq (6).
    end while
    Add $\{E, Y\}$ to $D_r^t$.
  end for
end for
```

2 Entity Classes for Each Step of Continual Few-Shot Learning

Table 1 and 2 shows the classes used for each step of continual few-shot learning. For CoNLL2003, we experiment with eights permutations with one class for each step. For OntoNote 5.0, we rank the 18 classes in alphabetic order and experiment with two combinations.

Table 1: Entity classes in each step with CoNLL2003.

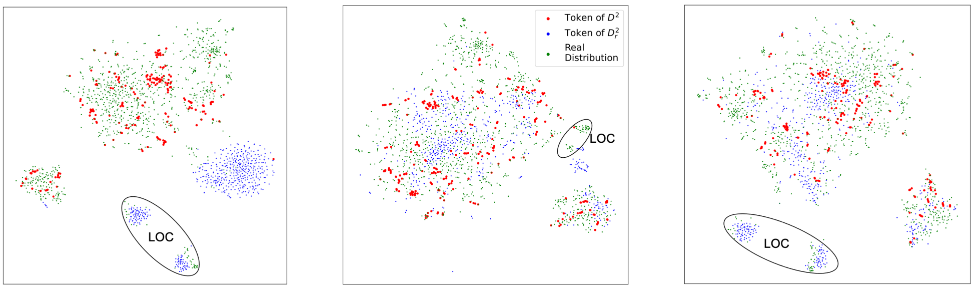
Class orderings for CoNLL2003
P_1: PER → LOC → ORG → MISC
P_2: PER → MISC → LOC → ORG
P_3: LOC → PER → ORG → MISC
P_4: LOC → ORG → MISC → PER
P_5: ORG → LOC → MISC → PER
P_6: ORG → MISC → PER → LOC
P_7: MISC → PER → LOC → ORG
P_8: MISC → ORG → PER → LOC

Table 2: Entity classes in each step with OntoNote 5.0. $[\cdot]$ means classes of the same step.

Class orderings for OntoNote 5.0
P_1: [CARDINAL, DATE, EVENT, FAC] → [GPE, LANGUAGE] → LAW → [LOC, MONEY] → NORP → [ORDINAL, ORG]→ PERCENT → [PERSON, PRODUCT] → [QUANTITY, TIME, WORK_OF_ART]
P_2: [CARDINAL, DATE, EVENT, FAC] → GPE → LANGUAGE→ LAW → LOC → [MONEY, NORP] → [ORDINAL, ORG] → [PERCENT, PERSON] → [PRODUCT, QUANTITY] → [TIME, WORK_OF_ART]

3 Hidden States of Tokens

Figure 3 shows the t-sne plots of hidden states of tokens from 10-shot *LOC→PER* (explained in the caption). In (a), we can find that there are synthetic tokens that are very close to the real *LOC* tokens (green dots in the black ellipse). These synthetic tokens (within the black ellipse) are the reconstructed *LOC*. On the contrary, the synthetic context, *i.e.*, the rest of the synthetic tokens outside the ellipse, are far away from the real distribution. This may because the context contains more diverse information, which makes it more difficult to be reconstructed. Such a difference between real and synthetic tokens may cause a domain shift between training and testing, since we are training on synthetic token and testing on real tokens. Note that there is no tokens from D^2 (red dots) in the black ellipse of *LOC*, indicating that there may not be *LOC* entities in the few-shot dataset D^2, unlike in continual learning where D^2 can contain a lot of entities of the old classes (*LOC*). (b) shows the result of matching all the synthetic tokens from D_r^2 with all the real ones from D^2. In this way, most of the synthetic tokens are matched with the real ones, except that only few synthetic tokens are aligned with the real *LOC* tokens. This is because the few-shot dataset D^2 may not contain entities from the old classes *LOC*. In this case, the adversarial matching will distract synthetic tokens from being reconstructed as *LOC*. Then, the reconstructed embedding sequences will contain less information from the old classes (*LOC*). In (c), we exclude synthetic tokens that are intended to be reconstructed as the old class *LOC*, *i.e.*, labeled as *LOC* in the target label sequence Y in Algorithm 1. As a result, the synthetic tokens contain both *LOC* and context that is aligned with the real distribution.



(a) (b) (c)

Figure 1: T-SNE plots of real and synthetic token embeddings with 10-shot *LOC→PER*, *i.e.*, training on *LOC* for step 1 and *PER* for step 2. D^2 is a 10-shot dataset for *PER*. We visualize the token embedding from the the last layer of the BERT encoder in M^1, *i.e.*, trained only on *LOC*. (a) *Ours ($\beta = 0$)*, no adversarial matching between tokens from D^2 and D_r^2. (b) *Ours (all tokens)*, matching all the tokens from D^2 and all the tokens from D_r^2. (c) *Ours*, excluding the synthetic tokens that are labeled as of entities from old classes, *i.e.*, *LOC*, from adversarial matching. The *real distribution* refers to tokens from the testing dataset, which are not available during training. We use black ellipses to mark tokens from the *real distribution* that are predicted as *LOC*, *i.e.*, the old entities.