A. Unfold Architecture of Figure 1 in the Main Paper

The unfold architecture of Figure 1 in the main paper is shown in Figure 1 of Appendix A.

![Figure 1. The unfold encoder-decoder framework.](image)

Z\_Y\_j for any j ∈ [T\_y] is calculated as:

\[ Z\_Y\_j = \frac{\sum_{i=1}^{T\_x} \alpha_i h\_Y\_j}{\sum_{i=1}^{T\_x} \exp(v^T \tanh(W_x h\_X\_i + W_y h\_Y\_i - 1))} \]  

(1)

where \( \alpha_i \) is calculated following (Bahdanau et al., 2015).

Z\_X\_i for any i ∈ [T\_x] is calculated as:

\[ Z\_X\_i = \frac{\sum_{j=1}^{T\_y} \beta_j h\_X\_i}{\sum_{j=1}^{T\_y} \exp(v^T \tanh(W_x h\_X\_i + W_y h\_Y\_j))} \]  

(2)

B. Unfold Architectures of X Component and Y Component in Figure 2 of the Main Paper

The unfold architectures of X Component and Y Component in Figure 2 of the main text is shown in Figure 2 of the appendix. Z\_X\_i and Z\_Y\_i are computed in the same ways as those in Eqn.(1) and Eqn.(2).

C. How to Build up the Dual Model

1. The Encoder: Set \( C_Y \) to the null context, i.e., \( C_Y = \{0\} \). At step j ∈ [T\_y] where T\_y is the length of y, preprocess \( C_Y \) and obtain Z\_Y\_j: Z\_Y\_j = \varphi_Y(h\_Y\_j - 1, C_Y). \( \varphi_Y \) is a function that sums up the elements in \( C_Y \) with adaptive weights. Then, calculate the hidden representation h\_Y\_j = \varphi_Y(y, h\_Y\_j - 1, Z\_Y\_j). Eventually, we obtain a set of hidden representations h\_Y = \{h\_Y\_j\}_{j=1}^{T\_y}. The module \( \varphi_X^i \) in component Y is not used while encoding y ∈ \( Y \).

2. The Decoder: Set \( C_X \) to the hidden representations h\_Y obtained in the encoding phase. At step i ∈ [T\_x], where T\_x is the length of x, preprocess \( C_X \) with the information available at step i and obtain Z\_X\_i: Z\_X\_i = \varphi_X(h\_X\_i - 1, C_X). Calculate the

\footnote{Note that in the encoding phase, all words in y are available. At step j, \( \varphi_Y \) and \( \varphi_Y^i \) can consider either y\_<j (Bahdanau et al., 2015) or all the y\_j’s (Vaswani et al., 2017).}
Figure 2. The unfold flow-chart of X component and Y component

hidden representation $h^X_i = \phi^X_c(x_{c,i}, h^X_{i-1}, Z^X_i)$. Then map $h^X_i$ to $x_i$ by $x_i = \phi^X_h(h^X_i)$. If $x_i$ is the symbol indicating the end of a sentence, terminate the decoding procedure; otherwise, continue to generate words one by one.

D. Theoretical Analysis

We give a brief theoretical discussion about model-level dual learning. Note that there are a primal model $f : \mathcal{X} \to \mathcal{Y}$ and a dual model $g : \mathcal{Y} \to \mathcal{X}$. The parameters of $f$ and $g$ are denoted as $\theta_f$ and $\theta_g$ respectively. We take the symmetric setting as an example and the result for the asymmetric setting is similarly obtained.

We want to minimize the (expected) risk of two models $f$ and $g$, which is defined as follows:

$$R(f,g) = \mathbb{E} \left[ \frac{1}{2} (\ell_1(f(x), y) + \ell_2(g(y), x)) \right],$$

$$\forall f \in \mathcal{F}, g \in \mathcal{G},$$

where $\mathcal{F} = \{ f(x; \theta_f); \theta_f \in \Theta_{xy} \}$, $\mathcal{G} = \{ g(y; \theta_g); \theta_g \in \Theta_{yx} \}$, $\Theta_{xy}$ and $\Theta_{yx}$ are parameter spaces, and the $\mathbb{E}$ is taken over the underlying data distribution $P$. The $\ell_1$ and $\ell_2$ in Eqn.(3) are loss functions, both of which are mappings $\mathcal{X} \times \mathcal{Y} \mapsto \mathbb{R}$.

As shown in Figure 1 of Section 1 at the main text, if we use two individual models to solve a pair of dual tasks, then for the primal task, we need to use a set of parameters $\phi^p_{c,Y}, \phi^p_{c,X}, \phi^p_{z,Y}, \phi^p_{h,Y}, \phi^p_{h,X}$, where the subscript $p$ stands for “primal”. The dual task needs another group of parameters $\phi^d_{c,Y}, \phi^d_{c,X}, \phi^d_{z,X}, \phi^d_{h,X}$, where the superscript $d$ stands for “dual”. By using our proposed method, we actually add the following constraints:

$$\phi^c_{p,Y} = \phi^c_{d,Y}; \quad \phi^c_{p,X} = \phi^c_{d,X}.$$  (4)

Let $\mathcal{T}$ denote the product space of the two models satisfying Eqn.(4). As a result, the model space of our proposed model-level dual learning is $(\mathcal{F} \times \mathcal{G}) \cap \mathcal{T}$, and we briefly denote it as $\mathcal{H}_1$.

Define the empirical risk on the $n$ sample as follows: for any $f \in \mathcal{F}, g \in \mathcal{G}$,

$$R_n(f,g) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{2n} (\ell_1(f(x_i), y_i) + \ell_2(g(y_i), x_i)).$$

$^2$The parameters $\theta_f$ and $\theta_g$ will be omitted when the context is clear.
Following (Bartlett & Mendelson, 2002), we introduce Rademacher complexity for our proposed method, a measure for the complexity of the hypothesis.

**Definition 1** Define the Rademacher complexity of our proposed method, $\mathcal{R}_n^d$, as follows:

$$
\mathcal{R}_n^d = \mathbb{E}_{z, \sigma} \left[ \sup_{(f, g) \in \mathcal{H}_1} \frac{1}{2n} \sum_{i=1}^{n} \sigma_i (\ell_1(f(x_i), y_i) + \ell_2(g(y_i), x_i)) \right],
$$

where $z = \{z_1, z_2, \ldots, z_n\} \sim P^n$, $z_i = (x_i, y_i)$ in which $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$, $\sigma = \{\sigma_1, \ldots, \sigma_m\}$ are i.i.d sampled with $P(\sigma_i = 1) = P(\sigma_i = -1) = 0.5$.

The following theorem generally holds for our proposed method:

**Theorem 1** (Theorem 3.1, (Mohri et al., 2012)) Let $\frac{1}{2} \ell_1(f(x), y) + \frac{1}{2} \ell_2(g(y), x)$ be a mapping from $\mathcal{X} \times \mathcal{Y}$ to $[0, 1]$. Then, for any $\delta \in (0, 1)$, with probability at least $1 - \delta$, the following inequality holds for any $(f, g) \in \mathcal{H}_1$,

$$
R(f, g) \leq R_n(f, g) + 2\mathcal{R}_n^d + \sqrt{\frac{1}{2n} \ln \left( \frac{1}{\delta} \right)}.
$$

Let $\mathcal{R}_n^c$ denote the Rademacher complexity for the standard supervised learning without our proposed method, i.e., no constraint like Eqn.(4) is applied. It is defined as follows:

**Definition 2** Define the Rademacher complexity of conventional learning scheme on the tasks $\mathcal{R}_n^c$, as follows:

$$
\mathcal{R}_n^c = \mathbb{E}_{z, \sigma} \left[ \sup_{(f, g) \in \mathcal{F} \times \mathcal{G}} \frac{1}{2n} \sum_{i=1}^{n} \sigma_i (\ell_1(f(x_i), y_i) + \ell_2(g(y_i), x_i)) \right],
$$

where $z = \{z_1, z_2, \ldots, z_n\} \sim P^n$, $z_i = (x_i, y_i)$ in which $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$, $\sigma = \{\sigma_1, \ldots, \sigma_m\}$ are i.i.d sampled with $P(\sigma_i = 1) = P(\sigma_i = -1) = 0.5$.

Considering $\mathcal{H}_1 \in \mathcal{F} \times \mathcal{G}$, by the definition of Rademacher complexity, we have $\mathcal{R}_n^d \leq \mathcal{R}_n^c$. Therefore, model-level dual learning has a smaller generation error bound than the conventional supervised learning.

**References**


