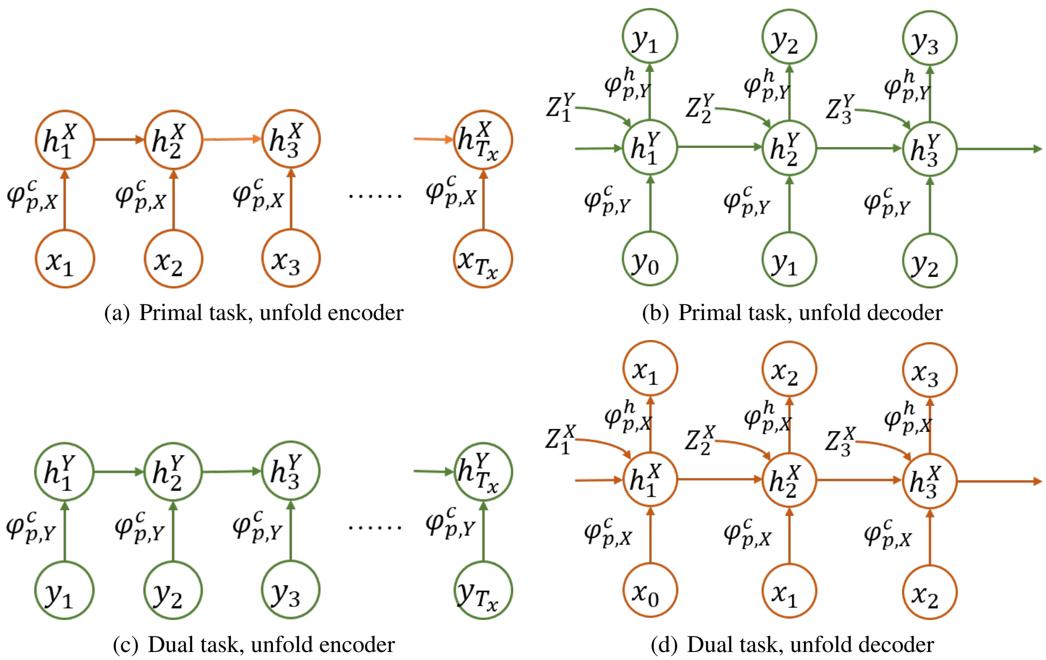
A. Unfold Architecture of Figure 1 in the Main Paper

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



(a) Primal task, unfold encoder (b) Primal task, unfold decoder

(c) Dual task, unfold encoder (d) Dual task, unfold decoder

Figure 1. The unfold encoder-decoder framework.

Z_j^Y for any $j \in [T_y]$ is calculated as:

$$Z_j^Y = \sum_{i=1}^{T_x} \alpha_i h_i^X, \ \alpha_i = \exp(v^T \tanh(W_x h_i^X + W_y h_{j-1}^Y)) / \sum_{i=1}^{T_x} \exp(v^T \tanh(W_x h_i^X + W_y h_{j-1}^Y)) \tag{1}$$

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

Z_i^X for any $i \in [T_x]$ is calculated as:

$$Z_i^X = \sum_{j=1}^{T_y} \beta_j h_j^Y, \ \beta_j = \exp(v^T \tanh(W_x h_{i-1}^X + W_y h_j^Y)) / \sum_{j=1}^{T_y} \exp(v^T \tanh(W_x h_{i-1}^X + W_y h_j^Y)). \tag{2}$$

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_j^X and Z_i^Y 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 $\mathcal{C}_Y$ to the null context, i.e., $\mathcal{C}_Y = \{0\}$. At step $j \in [T_y]$ where T_y is the length of y, preprocess $\mathcal{C}_Y$ and obtain Z_j^Y: $Z_j^X = \varphi_Y^z(h_{j-1}^Y, \mathcal{C}_Y)$. φ_Y^z is a function that sums up the elements in $\mathcal{C}_Y$ with adaptive weights. Then, calculate the hidden representation $h_j^Y = \varphi_Y^c(y, h_{j-1}^Y, Z_j^Y)$.[1] Eventually, we obtain a set of hidden representations $h^Y = \{h_j^Y\}_{j=1}^{T_y}$. The module φ_Y^h in component Y is not used while encoding $y \in \mathcal{Y}$.

(2) *The Decoder.* Set $\mathcal{C}_X$ to the hidden representations h^Y obtained in the encoding phase. At step $i \in [T_x]$, where T_x is the length of x, preprocess $\mathcal{C}_X$ with the information available at step i and obtain Z_i^X: $Z_i^X = \varphi_X^z(h_{i-1}^X, \mathcal{C}_X)$. Calculate the

[1]Note that in the encoding phase, all words in y are available. At step j, φ_Y^c and φ_Y^z can consider either $y_{<j}$ (Bahdanau et al., 2015) or all the y_j's (Vaswani et al., 2017).

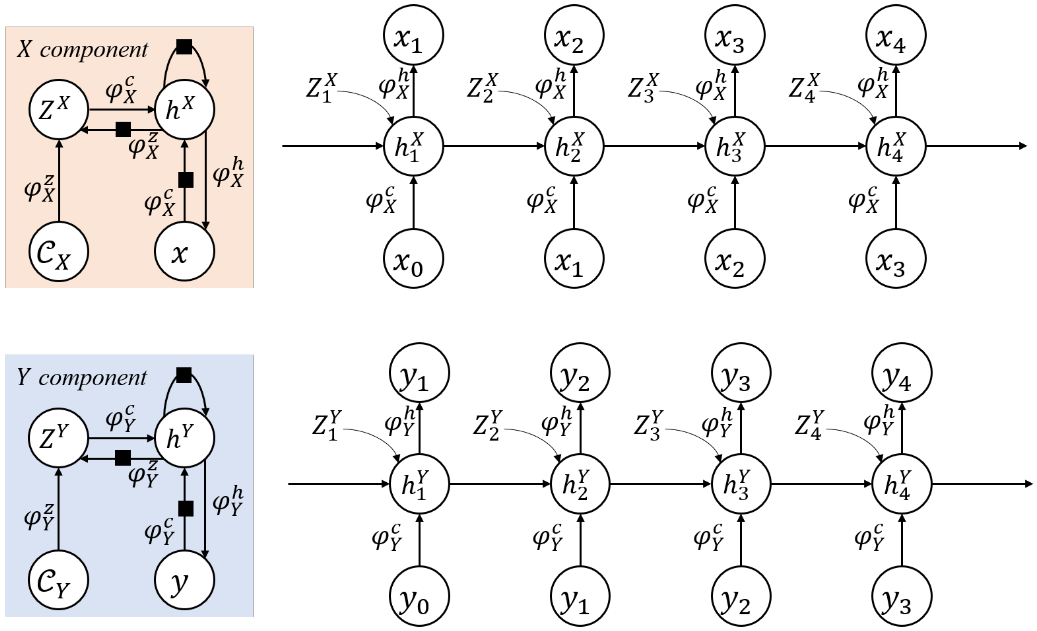


Figure 2. The unfold flow-chart of X component and Y component

hidden representation $h_i^X = \varphi_X^c(x_{<i}, h_{i-1}^X, Z_i^X)$. Then map h_i^X to x_i by $x_i = \varphi_X^h(h_j^X)$. 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 θ_f and θ_g respectively.[2] 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:

$$\begin{aligned} &R(f,g) = \mathbb{E}\left[\frac{1}{2}\big(\ell_1(f(x),y) + \ell_2(g(y),x)\big)\right], \\ &\forall f \in \mathcal{F}, g \in \mathcal{G}, \end{aligned} \tag{3}$$

where $\mathcal{F} = \{f(x;\theta_f); \theta_f \in \Theta_{xy}\}$, $\mathcal{G} = \{g(y;\theta_g); \theta_g \in \Theta_{yx}\}$, Θ_{xy} and Θ_{yx} are parameter spaces, and the $\mathbb{E}$ is taken over the underlying data distribution P. The ℓ_1 and ℓ_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 $\varphi_{p,X}^c, \varphi_{p,Y}^c, \varphi_{p,Y}^z, \varphi_{p,Y}^h$, where the subscript $_p$ stands for "primal". The dual task needs another group of parameters $\varphi_{d,Y}^c, \varphi_{d,X}^c, \varphi_{d,X}^z, \varphi_{d,X}^h$, where the superscript $_d$ stands for "dual". By using our proposed method, we actually add the following constraints:

$$\varphi_{p,Y}^c = \varphi_{d,Y}^c; \quad \varphi_{p,X}^c = \varphi_{d,X}^c. \tag{4}$$

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 θ_f and θ_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,* $\mathfrak{R}_n^d$*, as follows:*

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

where $\boldsymbol{z}=\{z_1,z_2,\cdots,z_n\}\sim P^n$*,* $z_i=(x_i,y_i)$ *in which* $x_i\in\mathcal{X}$ *and* $y_i\in\mathcal{Y}$*,* $\boldsymbol{\sigma}=\{\sigma_1,\cdots,\sigma_m\}$ *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* $(f,g)\in\mathcal{H}_1$*,*

$$R(f,g)\leq R_n(f,g)+2\mathfrak{R}_n^d+\sqrt{\frac{1}{2n}\ln(\frac{1}{\delta})}. \tag{5}$$

Let $\mathfrak{R}_n^c$ 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* $\mathfrak{R}_n^c$*, as follows:*

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

where $\boldsymbol{z}=\{z_1,z_2,\cdots,z_n\}\sim P^n$*,* $z_i=(x_i,y_i)$ *in which* $x_i\in\mathcal{X}$ *and* $y_i\in\mathcal{Y}$*,* $\boldsymbol{\sigma}=\{\sigma_1,\cdots,\sigma_m\}$ *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 $\mathfrak{R}_n^d\leq\mathfrak{R}_n^c$. Therefore, model-level dual learning has a smaller generation error bound than the conventional supervised learning.

References

Bahdanau, D., Cho, K., and Bengio, Y. Neural machine translation by jointly learning to align and translate. *ICLR*, 2015.

Bartlett, P. L. and Mendelson, S. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.

Mohri, M., Rostamizadeh, A., and Talwalkar, A. *Foundations of machine learning*. MIT press, 2012.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems*, pp. 6000–6010, 2017.