A. Justification of the Initialization in the Langevin Dynamics

In our approach, we leverage the Langevin dynamics to draw samples of the prototype vectors from their posterior distributions. Recall that the Langevin dynamics requires a burn-in period, which can take a long time. To accelerate the process, it is helpful to initialize the sample within a high-density area, which can prevent the sample from exploring low-density areas. Motivated by that, our approach aims to initialize the sample of prototype vector at a point with high posterior probability, and therefore we use the following strategy for initialization:

$$\hat{\mathbf{v}}_{\mathcal{T}} \leftarrow \{\hat{\mathbf{v}}_r\}_{r\in\mathcal{T}}, \text{ with each}$$
$$\hat{\mathbf{v}}_r \leftarrow \mathbf{m}_r + \mathbf{h}_r - \mathbf{m},$$

where $\mathbf{m}$ is the mean encoding of all the sentences in the support set, and $\mathbf{m}_r$ is the mean encoding for all the sentences of relation r in the support set. In the remainder of this section we justify this choice.

Our goal is to find the point with high posterior probability. Suppose we consider the N-way K-shot setting, where there are N relations in $\mathcal{T}$ (i.e., $|\mathcal{T}| = N$), and each relation has K examples in the support set. Then the posterior is given as:

$$\begin{aligned}
&\log p(\mathbf{v}_{\mathcal{T}}|\mathbf{x}_S, \mathbf{y}_S, \mathcal{G}) \\
=&\frac{1}{K}\log p(\mathbf{y}_S|\mathbf{x}_S, \mathbf{v}_{\mathcal{T}}) + \log p(\mathbf{v}_{\mathcal{T}}|\mathcal{G}) + \text{const} \\
=&\frac{1}{K}\sum_{s\in\mathcal{S}}\sum_{r\in\mathcal{T}}\mathbb{I}\{\mathbf{y}_s = r\}\log\frac{\exp(\mathcal{E}(\mathbf{x}_s)\cdot\mathbf{v}_r)}{\sum_{r'\in\mathcal{T}}\exp(\mathcal{E}(\mathbf{x}_s)\cdot\mathbf{v}_{r'})} + \sum_{r\in\mathcal{T}}\log\exp\left(-\frac{1}{2}\|\mathbf{v}_r - \mathbf{h}_r\|_2^2\right) + \text{const} \\
=&\frac{1}{K}\sum_{s\in\mathcal{S}}\sum_{r\in\mathcal{T}}\mathbb{I}\{\mathbf{y}_s = r\}(\mathcal{E}(\mathbf{x}_s)\cdot\mathbf{v}_r) - \frac{1}{K}\sum_{s\in\mathcal{S}}\log\sum_{r\in\mathcal{T}}\exp(\mathcal{E}(\mathbf{x}_s)\cdot\mathbf{v}_r) - \frac{1}{2}\sum_{r\in\mathcal{T}}\|\mathbf{v}_r - \mathbf{h}_r\|_2^2 + \text{const},
\end{aligned}$$

where we add a normalization term $\frac{1}{K}$ to the log-likelihood function, which makes the log-likelihood numerically stable as we increase the number of examples for each relation (i.e., K).

Our goal is to find a point of $\mathbf{v}_{\mathcal{T}}$ to maximize the above log-probability. However, the log-probability contains a log-partition function $\log\sum_{r\in\mathcal{T}}\exp(\mathcal{E}(\mathbf{x}_s)\cdot\mathbf{v}_r)$, which is hard to compute. To address this challenge, we aim at deriving a lower bound of the log-probability function for approximation. For this purpose, in this paper we use an inequation proposed by Simic (2008), which is formally stated in the following theorem.

Theorem A.1. *Suppose that $\tilde{x} = \{x_i\}_{i=1}^n$ represents a finite sequence of real numbers belonging to a fixed closed interval $I = [a, b]$, $a < b$. If f is a convex function on I, then we have that:*

$$\frac{1}{n}\sum_{i=1}^n f(x_i) - f\left(\frac{1}{n}\sum_{i=1}^n x_i\right) \le f(a) + f(b) - 2f\left(\frac{a+b}{2}\right).$$

Based on the above theorem, we can have the following corollary:

Corollary A.1. *Suppose that for all the $s \in \mathcal{S}$ and $r \in \mathcal{T}$, we have $\exp(\mathcal{E}(\mathbf{x}_s)\cdot\mathbf{v}_r) \in [a, b]$. As $(-\log)$ is a convex function, we therefore have:*

$$-\frac{1}{|\mathcal{T}|}\sum_{r\in\mathcal{T}}\log(\exp(\mathcal{E}(\mathbf{x}_s)\cdot\mathbf{v}_r)) + \log\left(\frac{1}{|\mathcal{T}|}\sum_{r\in\mathcal{T}}\exp(\mathcal{E}(\mathbf{x}_s)\cdot\mathbf{v}_r)\right) \le -\log(a) - \log(b) + 2\log\left(\frac{a+b}{2}\right).$$

After some simplification, we get:

$$\begin{aligned}
\log\left(\sum_{r\in\mathcal{T}}\exp(\mathcal{E}(\mathbf{x}_s)\cdot\mathbf{v}_r)\right) &\le \sum_{r\in\mathcal{T}}\frac{1}{|\mathcal{T}|}\log(\exp(\mathcal{E}(\mathbf{x}_s)\cdot\mathbf{v}_r)) + \log(|\mathcal{T}|) - \log(a) - \log(b) + 2\log\left(\frac{a+b}{2}\right) \\
&= \sum_{r\in\mathcal{T}}\frac{1}{|\mathcal{T}|}\mathcal{E}(\mathbf{x}_s)\cdot\mathbf{v}_r + \log(|\mathcal{T}|) - \log(a) - \log(b) + 2\log\left(\frac{a+b}{2}\right).
\end{aligned}$$

In practice, we can easily find such a and b so that $\exp(\mathcal{E}(\mathbf{x}_s)\cdot\mathbf{v}_r)\in[a,b]$ is satisfied. Given this corollary, we are able to obtain a lower bound of the log-proability function as follows:

$$
\begin{aligned}
&\log p(\mathbf{v}_{\mathcal{T}}|\mathbf{x}_S,\mathbf{y}_S,\mathcal{G})\\
=&\frac{1}{K}\sum_{s\in\mathcal{S}}\sum_{r\in\mathcal{T}}\mathbb{I}\{\mathbf{y}_s=r\}(\mathcal{E}(\mathbf{x}_s)\cdot\mathbf{v}_r)-\frac{1}{K}\sum_{s\in\mathcal{S}}\log\sum_{r\in\mathcal{T}}\exp(\mathcal{E}(\mathbf{x}_s)\cdot\mathbf{v}_r)-\frac{1}{2}\sum_{r\in\mathcal{T}}\|\mathbf{v}_r-\mathbf{h}_r\|_2^2+\text{const}\\
\geq&\frac{1}{K}\sum_{s\in\mathcal{S}}\sum_{r\in\mathcal{T}}\mathbb{I}\{\mathbf{y}_s=r\}(\mathcal{E}(\mathbf{x}_s)\cdot\mathbf{v}_r)-\frac{1}{K}\sum_{s\in\mathcal{S}}\left[\sum_{r\in\mathcal{T}}\frac{1}{|\mathcal{T}|}\mathcal{E}(\mathbf{x}_s)\cdot\mathbf{v}_r+\log(|\mathcal{T}|)-\log(a)-\log(b)+2\log\left(\frac{a+b}{2}\right)\right]\\
&-\frac{1}{2}\sum_{r\in\mathcal{T}}\|\mathbf{v}_r-\mathbf{h}_r\|_2^2+\text{const}\\
=&\frac{1}{K}\sum_{s\in\mathcal{S}}\sum_{r\in\mathcal{T}}\mathbb{I}\{\mathbf{y}_s=r\}(\mathcal{E}(\mathbf{x}_s)\cdot\mathbf{v}_r)-\frac{1}{K}\sum_{s\in\mathcal{S}}\sum_{r\in\mathcal{T}}\frac{1}{N}\mathcal{E}(\mathbf{x}_s)\cdot\mathbf{v}_r-\frac{1}{2}\sum_{r\in\mathcal{T}}\|\mathbf{v}_r-\mathbf{h}_r\|_2^2+\text{const}.
\end{aligned}
$$

Based on that, let us denote $\mathbf{m}=\frac{1}{NK}\sum_{s\in\mathcal{S}}\mathcal{E}(\mathbf{x}_s)$ to be the mean of encodings for all the sentences in the support set, and denote $\mathbf{m}_r=\frac{1}{K}\sum_{s\in\mathcal{S}}\mathbb{I}\{\mathbf{y}_s=r\}\mathcal{E}(\mathbf{x}_s)$ to be the mean of encodings for sentences under relation r in the support set. In this way, the above lower bound can be rewritten as follows:

$$
\begin{aligned}
&\log p(\mathbf{v}_{\mathcal{T}}|\mathbf{x}_S,\mathbf{y}_S,\mathcal{G})\\
\geq&\frac{1}{K}\sum_{s\in\mathcal{S}}\sum_{r\in\mathcal{T}}\mathbb{I}\{\mathbf{y}_s=r\}(\mathcal{E}(\mathbf{x}_s)\cdot\mathbf{v}_r)-\frac{1}{K}\sum_{s\in\mathcal{S}}\sum_{r\in\mathcal{T}}\frac{1}{N}\mathcal{E}(\mathbf{x}_s)\cdot\mathbf{v}_r-\frac{1}{2}\sum_{r\in\mathcal{T}}\|\mathbf{v}_r-\mathbf{h}_r\|_2^2+\text{const}\\
=&\sum_{r\in\mathcal{T}}\left[\mathbf{v}_r\cdot\mathbf{m}_r-\mathbf{v}_r\cdot\mathbf{m}+\mathbf{v}_r\cdot\mathbf{h}_r-\frac{1}{2}\mathbf{v}_r\cdot\mathbf{v}_r-\frac{1}{2}\mathbf{h}_r\cdot\mathbf{h}_r\right]+\text{const}\\
=&\sum_{r\in\mathcal{T}}\left[\mathbf{v}_r\cdot\mathbf{m}_r-\mathbf{v}_r\cdot\mathbf{m}+\mathbf{v}_r\cdot\mathbf{h}_r-\frac{1}{2}\mathbf{v}_r\cdot\mathbf{v}_r\right]+\text{const}\\
=&\sum_{r\in\mathcal{T}}\left[-\frac{1}{2}\|\mathbf{v}_r-\mathbf{m}_r-\mathbf{h}_r+\mathbf{m}\|_2^2\right]+\text{const}.
\end{aligned}
$$

Therefore, under this lower bound, the optimal initialization of prototype vector $\mathbf{v}_r$ for each relation $r\in\mathcal{T}$ is given by:

$$\mathbf{v}_r^*=\mathbf{m}_r+\mathbf{h}_r-\mathbf{m},$$

where we can ensure $\mathbf{v}_r^*$ to have a pretty high probability under the posterior distribution, and hence the Langevin dynamics is likely to converge faster.

B. Details on the Sentence Encoder

In our approach, we use the entity marker method proposed in Soares et al. (2019) to generate the encoding of each sentence with $\text{BERT}_{\texttt{BASE}}$ (Devlin et al., 2019). More specifically, recall that the goal of relation extraction is to predict the relation between two entities expressed in a sentence. Therefore, each sentence contains two entity mentions, i.e., token spans corresponding to an entity. For example, a sentence can be "**Washington** is the capital of the **United States** .", where **Washington** and **United States** are the entity mentions. During preprocessing, we follow Soares et al. (2019) and add two markers for each entity mention, including a starting marker before the entity mention and an ending marker after the entity mention. In this way, the example sentence becomes "*[E1]* **Washington** *[/E1]* is the capital of the *[E2]* **United States** *[/E2]* .". Here, *[E1]* and *[E2]* are the starting markers. *[/E1]* and *[/E2]* are the ending markers. Then we apply $\text{BERT}_{\texttt{BASE}}$ to the preprocessed sentence, yielding an embedding vector for each token in the sentence. Finally, we follow Soares et al. (2019) to concatenate the embeddings of token *[E1]* and token *[E2]* as the sentence encoding.

C. Comparison of Similarity Measures

In our approach, given the encoding $\mathcal{E}(\mathbf{x})$ of a sentence $\mathbf{x}$ and relation prototype vectors $\mathbf{v}_{\mathcal{T}}$, we predict the label $\mathbf{y}$ as below:

$$p(\mathbf{y}=r|\mathbf{x},\mathbf{v}_{\mathcal{T}})=\frac{\exp(\mathcal{E}(\mathbf{x})\cdot\mathbf{v}_r)}{\sum_{r'\in\mathcal{T}}\exp(\mathcal{E}(\mathbf{x})\cdot\mathbf{v}_{r'})},$$

where we compute the dot product between sentence encodings and relation prototype vectors, and treat the value as logits for classification. Intuitively, the dot product could be understood as a similarity measure between encodings and prototype vectors. Besides dot product, Euclidean distance is another widely-used similarity measure, and we could naturally change the similarity measure in our approach to Euclidean distance as follows:

$$p(\mathbf{y} = r|\mathbf{x}, \mathbf{v}_{\mathcal{T}}) = \frac{\exp(-\frac{1}{2}\left\|\mathcal{E}(\mathbf{x}) - \mathbf{v}_r\right\|^2)}{\sum_{r' \in \mathcal{T}} \exp(-\frac{1}{2}\left\|\mathcal{E}(\mathbf{x}) - \mathbf{v}_{r'}\right\|^2)}.$$

In this section, we empirically compare the results of the two similarity measures, where the same configuration of hyperparameters is used for both similarity measures. The results are presented in Tab. 1 and Tab. 2, where we can see that dot product works better in the 1-shot learning setting, whereas Euclidean distance achieves higher accuracy in the 5-shot learning setting. Therefore, when the number of support sentences for each relation is very limited (e.g., 1 or 2), it is better to use dot product. When we have more support sentences (e.g., 5 or more per relation), Euclidean distance is a better choice.

Table 1. Results on FewRel test set.

Similarity Measure	**5-Way 1-Shot**	**5-Way 5-Shot**	**10-Way 1-Shot**	**10-Way 5-Shot**
Dot Product	**90.30**	94.25	**84.09**	**89.93**
Euclidean Distance	86.74	**94.34**	78.56	88.95

Table 2. Results on FewRel validation set.

Similarity Measure	**5-Way 1-Shot**	**5-Way 5-Shot**	**10-Way 1-Shot**	**10-Way 5-Shot**
Dot Product	**87.95**	92.54	**80.26**	86.72
Euclidean Distance	86.79	**94.44**	78.48	**88.92**

References

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019.

Simic, S. On a global upper bound for jensen's inequality. *Journal of mathematical analysis and applications*, 2008.

Soares, L. B., FitzGerald, N., Ling, J., and Kwiatkowski, T. Matching the blanks: Distributional similarity for relation learning. In *ACL*, 2019.