Acknowledgments

We thank Yizhe Zhang, Sungjin Lee, Dinghan Shen, Wenlin Wang for insightful discussion. The implementation in our experiments heavily depends on three NLP applications published on Github repositories; we acknowledge all the authors who made their code public, which tremendously accelerates our project progress.

A Comparison of different schedules

We compare the two different scheduling schemes in Figure 7. The three widely used monotonic schedules are shown in the top row, including linear, sigmoid and cosine. We can easily turn them into their corresponding cyclical versions, shown in the bottom row.

B Proofs on the β and MI

When scheduled with β, the training objective over the dataset can be written as:

$$\mathcal{F} = -\mathcal{F}_E + \beta \mathcal{F}_R \tag{12}$$

We proceed the proof by re-writing each term separately.

B.1 Bound on $\mathcal{F}_E$

Following (Li et al., 2017), on the support of $(\boldsymbol{x}, \boldsymbol{z})$, we denote q as the encoder probability measure, and p as the decoder probability measure. Note that the reconstruction loss for $\boldsymbol{z}$ can be writen as its negative log likelihood form as:

$$\mathcal{F}_E = -\mathbb{E}_{\boldsymbol{x}\sim q(\boldsymbol{x}), \boldsymbol{z}\sim q(\boldsymbol{z}|\boldsymbol{x})}[\log p(\boldsymbol{x}|\boldsymbol{z})]. \tag{13}$$

Lemma 1 *For random variables $\boldsymbol{x}$ and $\boldsymbol{z}$ with two different probability measures, $p(\boldsymbol{x}, \boldsymbol{z})$ and $q(\boldsymbol{x}, \boldsymbol{z})$, we have*

$$\begin{aligned}
&H_p(\boldsymbol{z}|\boldsymbol{x}) \\
&= -\mathbb{E}_{\boldsymbol{z}\sim p(\boldsymbol{z}), x\sim p(\boldsymbol{x}|\boldsymbol{z})}[\log p(\boldsymbol{z}|\boldsymbol{x})] \\
&= -\mathbb{E}_{\boldsymbol{z}\sim p(\boldsymbol{z}), \boldsymbol{x}\sim p(\boldsymbol{x}|\boldsymbol{z})}[\log q(\boldsymbol{z}|\boldsymbol{x})] \\
&- \mathbb{E}_{\boldsymbol{z}\sim p(\boldsymbol{z}), \boldsymbol{x}\sim p(\boldsymbol{x}|\boldsymbol{z})}\big[\log p(\boldsymbol{z}|\boldsymbol{x}) - \log q(\boldsymbol{z}|\boldsymbol{x})\big] \\
&= -\mathbb{E}_{\boldsymbol{z}\sim p(\boldsymbol{z}), \boldsymbol{x}\sim p(\boldsymbol{x}|\boldsymbol{z})}[\log q(\boldsymbol{z}|\boldsymbol{x})] \\
&- \mathbb{E}_{p(\boldsymbol{x})}(KL(p(\boldsymbol{z}|\boldsymbol{x})||q(\boldsymbol{z}|\boldsymbol{x}))) \\
&\leq -\mathbb{E}_{\boldsymbol{z}\sim p(\boldsymbol{z}), \boldsymbol{x}\sim p(\boldsymbol{x}|\boldsymbol{z})}[\log q(\boldsymbol{z}|\boldsymbol{x})] \qquad (14)
\end{aligned}$$

where $H_p(\boldsymbol{z}|\boldsymbol{x})$ is the conditional entropy. Similarly, we can prove that

$$H_q(\boldsymbol{x}|\boldsymbol{z}) \leq -\mathbb{E}_{\boldsymbol{x}\sim q(\boldsymbol{x}), \boldsymbol{z}\sim q(\boldsymbol{z}|\boldsymbol{x})}[\log p(\boldsymbol{x}|\boldsymbol{z})] \tag{15}$$

From lemma 1, we have

Corollary 1 *For random variables $\boldsymbol{x}$ and $\boldsymbol{z}$ with probability measure $p(\boldsymbol{x}, \boldsymbol{z})$, the mutual information between $\boldsymbol{x}$ and $\boldsymbol{z}$ can be written as*

$$\begin{aligned}
I_q(\boldsymbol{x}, \boldsymbol{z}) &= H_q(\boldsymbol{x}) - H_q(\boldsymbol{x}|\boldsymbol{z}) \geq H_q(\boldsymbol{x}) \\
&+ \mathbb{E}_{\boldsymbol{x}\sim q(\boldsymbol{x}), \boldsymbol{z}\sim q(\boldsymbol{z}|\boldsymbol{x})}[\log p(\boldsymbol{x}|\boldsymbol{z})] \\
&= H_q(\boldsymbol{x}) + \mathcal{F}_E \qquad (16)
\end{aligned}$$

B.2 Decomposition of $\mathcal{F}_R$

The KL term in (5) can be decomposed into two refined terms (Hoffman and Johnson, 2016):

$$\begin{aligned}
&\mathcal{F}_R \\
&= \mathbb{E}_{q(\boldsymbol{x})}[\mathrm{KL}(q(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z}))] \\
&= \mathbb{E}_{q(\boldsymbol{z},\boldsymbol{x})}[(\log q(\boldsymbol{z}|\boldsymbol{x}) - \log p(\boldsymbol{z}))] \\
&= \mathbb{E}_{q(\boldsymbol{z},\boldsymbol{x})}[(\log q(\boldsymbol{z}|\boldsymbol{x}) - \log q(\boldsymbol{z})] \\
&\quad + \mathbb{E}_{q(\boldsymbol{z},\boldsymbol{x})}[\log q(\boldsymbol{z}) - \log p(\boldsymbol{z}))] \\
&= \mathbb{E}_{q(\boldsymbol{z},\boldsymbol{x})}[(\log q(\boldsymbol{z},\boldsymbol{x}) - \log q(\boldsymbol{x}) - \log q(\boldsymbol{z})] \\
&\quad + \mathbb{E}_{q(\boldsymbol{z},\boldsymbol{x})}[\log q(\boldsymbol{z}) - \log p(\boldsymbol{z}))] \\
&= \underbrace{I_q(\boldsymbol{z}, \boldsymbol{x})}_{\mathcal{F}_1\text{: Mutual Info.}} + \underbrace{\mathrm{KL}(q(\boldsymbol{z})||p(\boldsymbol{z}))}_{\mathcal{F}_2\text{: Marginal KL}} \qquad (17)
\end{aligned}$$

C Model Description

C.1 Conditional VAE for dialog

Each conversation can be represented via three random variables: the dialog context $\boldsymbol{c}$ composed of the dialog history, the response utterance $\boldsymbol{x}$, and a latent variable $\boldsymbol{z}$, which is used to capture the latent distribution over the valid responses($\beta = 1$) (Zhao et al., 2017). The ELBO can be written as:

$$\begin{aligned}
\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{c}) &\geq \mathcal{L}_{\text{ELBO}} \qquad (18) \\
&= \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{c})}\big[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{c})\big] \\
&- \beta \mathrm{KL}(q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{c})||p(\boldsymbol{z}|\boldsymbol{c}))
\end{aligned}$$

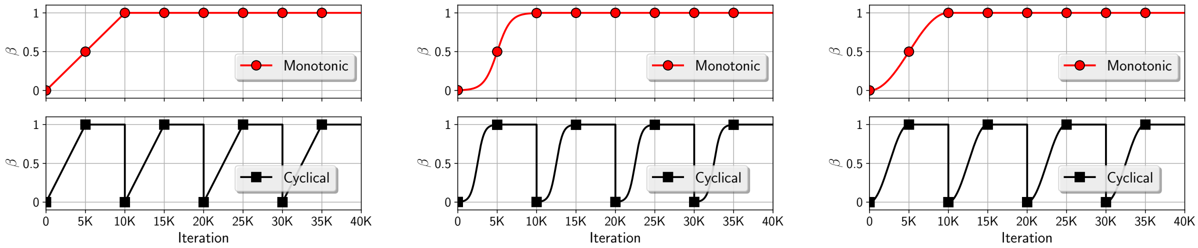
C.2 Semi-supervised learning with VAE

We use a simple factorization to derive the ELBO for semi-supervised learning. α is introduced to regularize the strength of classification loss.

$$\begin{aligned}
&\log p_{\boldsymbol{\theta}}(\boldsymbol{y}, \boldsymbol{x}) \geq \mathcal{L}_{\text{ELBO}} \qquad (19) \\
&= \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\big[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}) + \alpha \log p_{\boldsymbol{\psi}}(\boldsymbol{y}|\boldsymbol{z})\big] \\
&- \beta \mathrm{KL}(q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z})) \qquad (20)
\end{aligned}$$

where $\boldsymbol{\psi}$ is the parameters for the classifier.

Good latent codes $\boldsymbol{z}$ are crucial for the the classification performance, especially when simple classifiers are employed, or less labelled data is available.



(a) Linear (b) Sigmoid (c) Cosine

Figure 7: Comparison between traditional monotonic and proposed cyclical annealing schedules. The top row shows the traditional monotonic schedules, and the bottom row shows their corresponding cyclical schedules. $M = 4$ cycles are illustrated, $R = 0.5$ is used for annealing within each cycle.

D More Experimental Results

D.1 CVAE for Dialog Response Generation

Code & Dataset We implemented different schedules based on the code[5] published by Zhao et al. (2017). In the SW dataset, there are 70 available topics. We randomly split the data into 2316/60/62 dialogs for train/validate/test.

Results The results on full BLEU scores are shown in Table 5. The cyclical schedule outperforms the monotonic schedule in both settings. The learning curves are shown in Figure 8. Under similar ELBO results, the cyclical schedule provide lower reconstruction errors, higher KL values, and higher BLEU values than the monotonic schedule. Interestingly, the monotonic schedule tends to overfit, while the cyclical schedule does not, particularly on reconstruction errors. It means the monotonic schedule can learn better latent codes for VAEs, thus preventing overfitting.

D.2 Semi-supervised Text Classification

Dataset Yelp restaurant reviews dataset utilizes user ratings associated with each review. Reviews with rating above three are considered positive, and those below three are considered negative. Hence, this is a binary classification problem. The pre-processing in (Shen et al., 2017) allows sentiment analysis on sentence level. It further filters the sentences by eliminating those that exceed 15 words. The resulting dataset has 250K negative sentences, and 350K positive ones. The vocabulary size is 10K after replacing words occurring less than 5 times with the "<unk>" token.

Results The tSNE embeddings are visualized in Figure 9. We see that cyclical β provides much more separated latent structures than the other two methods.

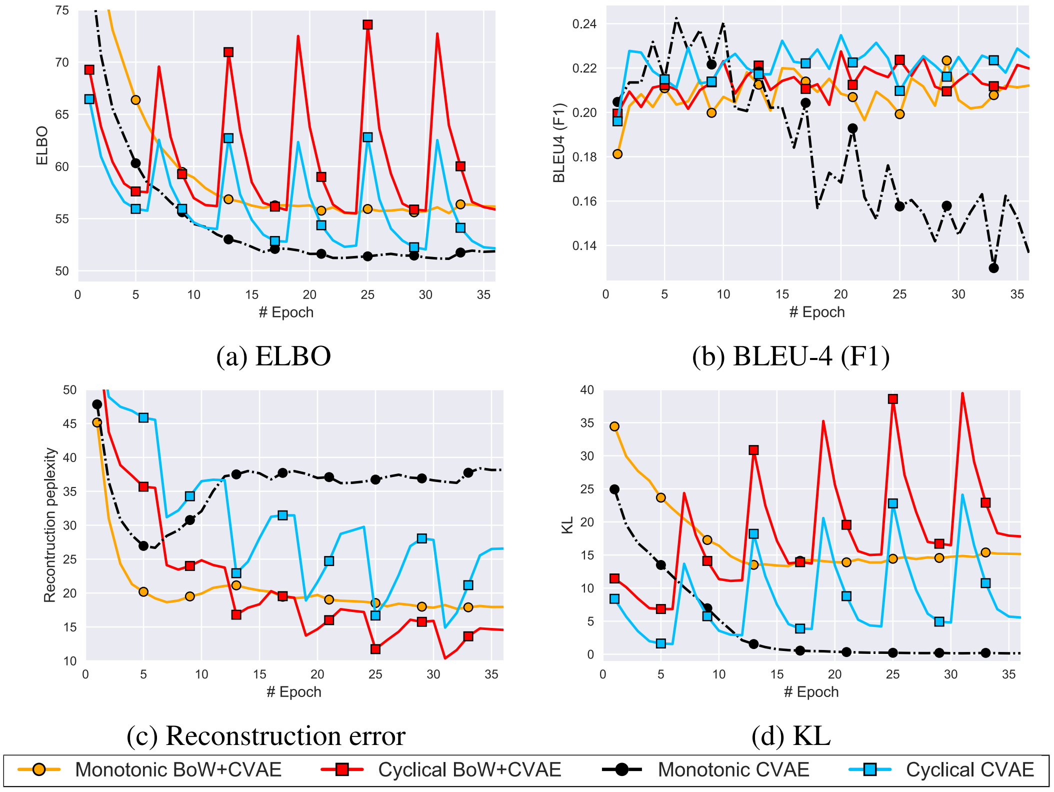
Model	CVAE		CVAE+BoW	
Schedule	**M**	**C**	**M**	**C**
B1 prec	0.326	**0.423**	0.384	**0.397**
B1 recall	0.214	**0.391**	0.376	**0.387**
B2 prec	0.278	**0.354**	0.320	**0.331**
B2 recall	0.180	**0.327**	0.312	**0.323**
B3 prec	0.237	**0.299**	0.269	**0.279**
B3 recall	0.153	**0.278**	0.265	**0.275**
B4 prec	0.185	**0.234**	0.211	**0.219**
B4 recall	0.122	**0.220**	0.210	**0.219**

Table 5: Comparison on dialog response generation. BLEU (B) scores 1-4 are used for evaluation. Monotonic (**M**) and Cyclical (**C**) schedules are tested on two models.

D.3 Hyper-parameter tuning

The cyclical schedule has two hyper-parameters M and R. We provide the full results on M and R in Figure 11 and Figure 12, respectively. A larger number of cycles M can provide higher performance for various proportion value R.

[5] https://github.com/snakeztc/NeuralDialog-CVAE



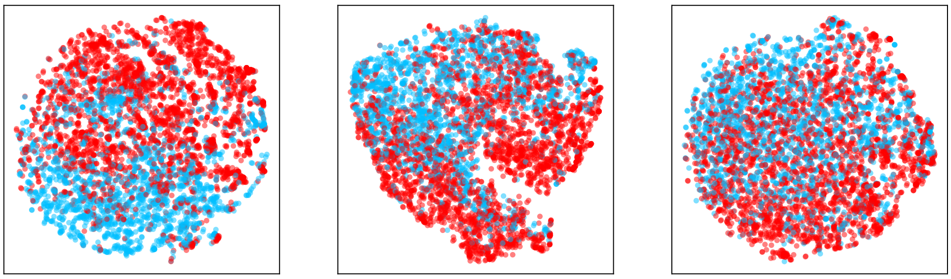
(a) ELBO

(b) BLEU-4 (F1)

(c) Reconstruction error

(d) KL

Figure 8: Full results of CVAE and BoW+CVAE on SW dataset. Under similar ELBO results, the cyclical schedule provide lower reconstruction errors, higher KL values, and higher BLEU values than the monotonic schedule. Interestingly, the monotonic schedule tends to overfit, while the cyclical schedule does not.

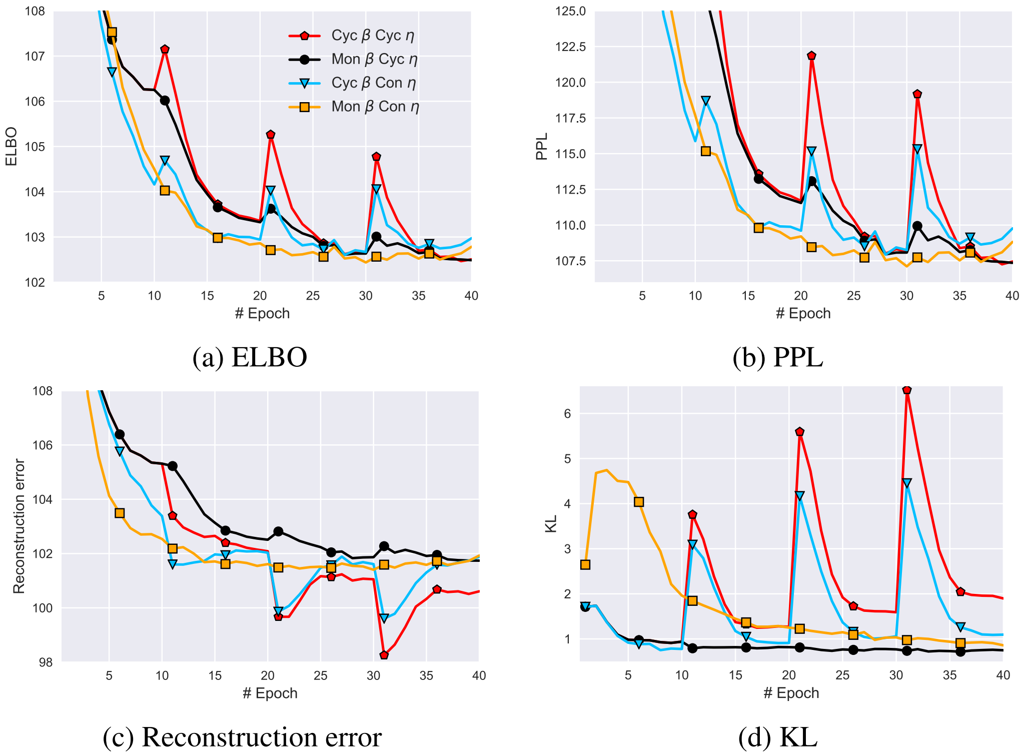


(a) Cyclical β VAE

(b) Monotonic β VAE

(c) AE

Figure 9: Comparison of tSNE embeddings for three methods on Yelp dataset. This can be considered as the unsupervised feature learning results in semi-supervised learning. More structured latent patterns usually lead to better classification performance.



(a) ELBO

(b) PPL

(c) Reconstruction error

(d) KL

Figure 10: Ablation study on cyclical schedules on β and η.

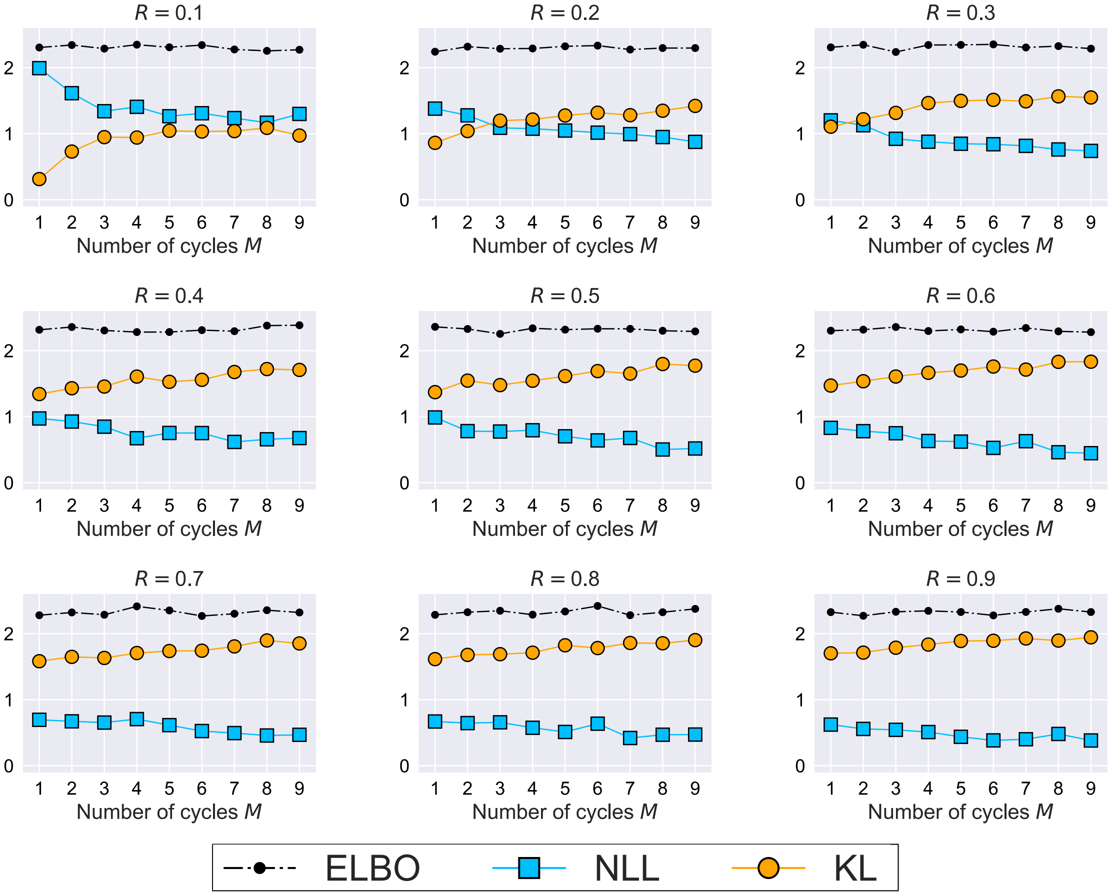


Figure 11: The impact of hyper-parameter M: number of cycles. A larger number of cycles lead to better performance. The improvement is more significant when R is small. The improvement is small when R is large.

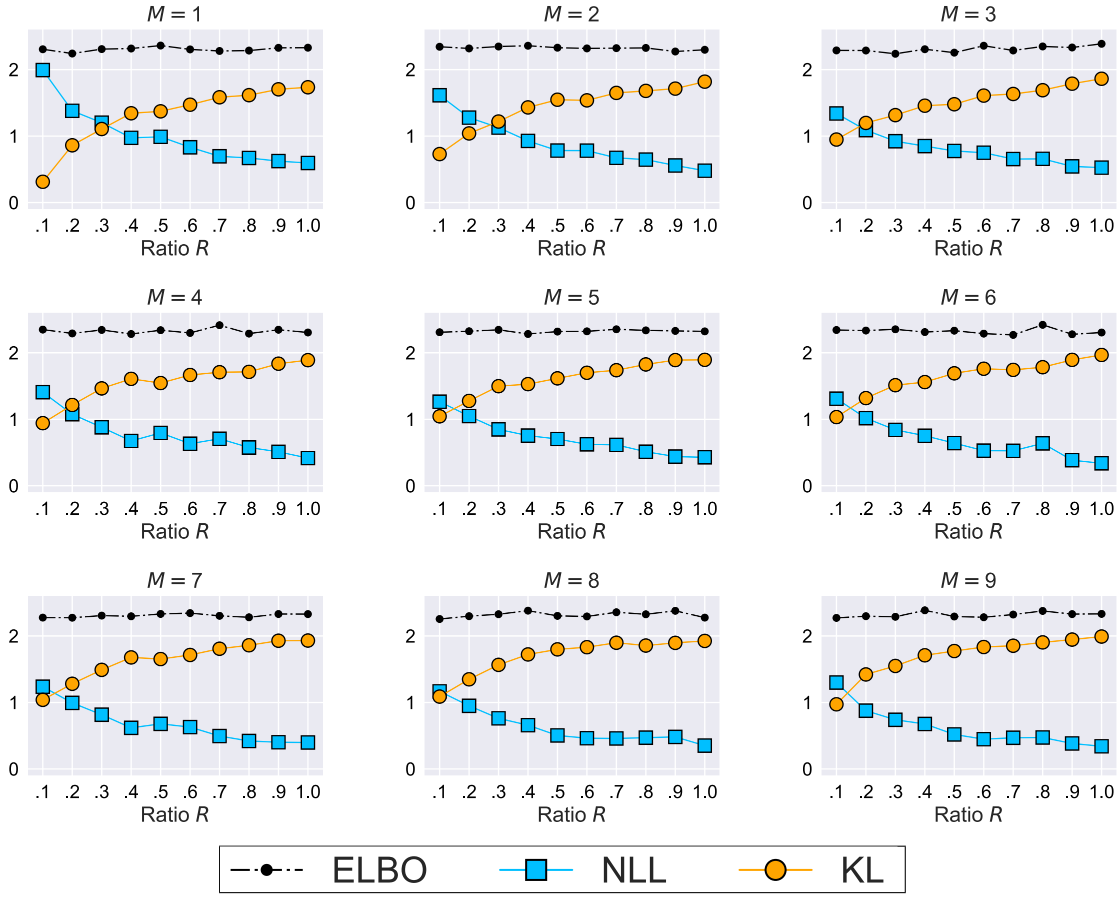


Figure 12: The impact of hyper-parameter R: proportion for the annealing stage. A larger R leads to better performance for various M. Small R performs worse, because the schedule becomes more similar with constant schedule. $M = 1$ recovers the monotonic schedule. Contrary to the convention that typically adopts small R, our results suggests that larger R should be considered.