Taming Continuous Posteriors for Latent Variational Dialogue Policies

Marin Vlastelica[1], **Patrick Ernst**[2], **György Szarvas**[2]

[1]Autonomous Learning Group, Max Planck Institute for Intelligent Systems, Tübingen, Germany
[2]Amazon Development Center Germany GmbH, Berlin, Germany
[1]marin.vlastelica@tuebingen.mpg.de, [2]{peernst, szarvasg}@amazon.de

Abstract

Utilizing amortized variational inference for latent-action reinforcement learning (RL) has been shown to be an effective approach in Task-oriented Dialogue (ToD) systems for optimizing dialogue success. Until now, categorical posteriors have been argued to be one of the main drivers of performance. In this work we revisit Gaussian variational posteriors for latent-action RL and show that they can yield even better performance than categoricals. We achieve this by introducing an improved variational inference objective for learning continuous representations without auxiliary learning objectives, which streamlines the training procedure. Moreover, we propose ways to regularize the latent dialogue policy, which helps to retain good response coherence. Using continuous latent representations our model achieves state of the art dialogue success rate on the MultiWOZ benchmark, and also compares well to categorical latent methods in response coherence.

1 Introduction

Task-oriented Dialogue (ToD) systems have reached a degree of maturity, which enables them to engage with human users and assist them in various tasks. They are able to steer natural-language conversations in order to complete users' goals, such as booking restaurants, querying weather forecasts and resolving customer service issues. At their core, the behavior of these systems is controlled by a dialogue policy, which receives user inputs in the form of utterances and additional features or states.

Template-based methods (Walker et al. 2007; Inaba and Takahashi 2016) leverage ranking or classification approaches to select the most fitting response from a pre-defined set of responses, i.e. templates. While template-based methods offer better control over the dialogue policy behavior, they are less versatile due to their dependency on template sets. Moreover, constructing comprehensive template sets is a challenge in itself (Gao, Galley, and Li 2019). In retrieval-based approaches (Yan, Song, and Wu 2016; Henderson et al. 2019; Tao et al. 2019) candidate responses are not pre-defined, but are retrieved from massive dialogue corpora, e.g. by executing ad-hoc search queries a priori.

Generative models do not require such additional inputs as prior knowledge. They enable end-to-end (E2E) learning of dialogue policies and have the potential to generate diverse responses by leveraging a large vocabulary (Serban et al. 2017; Zhao, Zhao, and Eskenazi 2017; Gu et al. 2018). Even though such fully data-driven approaches offer great versatility and a faster adoption, they may exhibit degenerate behavior by generating incomprehensible utterances. This is apparent in multi-turn dialogues, which span hundreds of words, while the success signal is only observed at the end of dialogues. Reinforcement learning (RL) based approaches are able to optimize for such long-term, sparse rewards and have been applied in this setup. Previous approaches prominently applied word-level RL (Lewis et al. 2017; Kottur et al. 2017), where the action space is defined over the entire vocabulary. The response utterances are then generated auto-regressively by consecutive next-word predictions. Unfortunately, the use of large action spaces often impedes the convergence of policy learning algorithms, which makes it hard to ensure coherent responses. Prior work makes use of *latent-action RL* to address the dimensionality problem by utilizing variational inference approaches (Zhao, Xie, and Eskenazi 2019; Lubis et al. 2020). These methods rely on a supervised learning stage, where latent action representations are learned over response utterances, followed by fine-tuning via reinforcement learning in the latent space.

In this paper we follow this paradigm, but extend prior work substantially by introducing the TCUP approach, which aims to *T*ame *C*ontin*U*ous *P*osteriors for latent variational dialogue policies. TCUP makes the following contributions: **(i)** A new formulation of the variational inference objective for learning continuous latent response representations without auxiliary learning objectives. **(ii)** A more robust approach for learning from ToD data in an offline RL setup, which utilizes the fact that we are dealing with expert dialogue trajectories.

Using the MultiWOZ benchmark (Budzianowski et al. 2018), we show that TCUP is able to improve the state-of-the-art performance across different metrics. In addition to MultiWOZ's context-to-text metrics and following Lubis et al. (2020), we demonstrate the benefits of the learned continuous latent representations quantitatively, using a clustering analysis.

2 Preliminaries

Latent-action reinforcement learning methods are trained in two stages: In the first stage, which we denote as *SL Stage*,

an encoder-decoder architecture is trained using a supervised approach. Leveraging a corpus of real dialogues, the encoder learns a latent representation over dialogue histories and responses; the decoder network then learns to generate the reference responses constrained to the encoder's representation as input. The learned representations compress the dialogue response space and serve as latent actions. In a consecutive stage, denoted as *RL Stage*, the encoder is further finetuned by a RL approach to improve the latent action predictions, optimizing the long-term dialogue success. In the *RL Stage* the decoder stays fixed and receives the output of the RL policy to generate the final response utterance.

2.1 *SL Stage* – Learning Latent Response Representations

We denote a dialogue context as c, which contains the dialogue history and state. Response utterances are denoted as x. Provided a dataset of context and optimal response pairs (c, x), we want to extract a latent representation z, representing the dialogue responses given context. Approaches based on variational inference have shown to be beneficial for learning such latent representations. This is done by optimizing the evidence lower bound (ELBO)

$$\begin{aligned}\mathcal{L}(\phi, \theta) = \mathbb{E}_{q_\theta(\mathrm{z}|\mathrm{x},\mathrm{c})}[-\log p_\phi(\mathrm{x}|\mathrm{z})] \\ + D_{\mathrm{KL}}[q_\theta(\mathrm{z}|\mathrm{x},\mathrm{c})||p(\mathrm{z}|\mathrm{c})],\end{aligned} \tag{1}$$

where q_θ denotes the variational posterior parameterized by θ and p_ϕ the decoder parameterized by ϕ. Prior work (Zhao, Xie, and Eskenazi 2019) argues that this full ELBO formulation suffers from "explaining away", i.e. without additional incentive, the encoder only relies on x for computing z. To mitigate this overexposure bias to responses, i.e. the insensitivity to the context, the authors introduce the "lite" ELBO:

$$\begin{aligned}\mathcal{L}(\phi, \theta) = \mathbb{E}_{q_\theta(\mathrm{z}|\mathrm{c})}[-\log p_\phi(\mathrm{x}|\mathrm{z})] \\ + D_{\mathrm{KL}}[q_\theta(\mathrm{z}|\mathrm{c})||p(\mathrm{z})].\end{aligned} \tag{2}$$

The goal is to be closer to the information available during testing time, where the decoder only sees z conditionally sampled on the context $q_\theta(\mathrm{z}|\mathrm{c})$.

2.2 *RL Stage* – Learning Latent Dialogue Policies

After learning to extract a compressed representation z in the supervised learning stage, the encoder $q_\phi(\mathrm{z}|\mathrm{c})$ is finetuned via reinforcement learning to optimize the dialogue reward, which is typically based on a dialogue success / completion signal. A Markov Decision Process ($\mathcal{MDP}$) is defined as a tuple $(\mathcal{S}, \mathcal{A}, r, p)$ of state space $\mathcal{S}$, action space $\mathcal{A}$, reward function r, and transition density p. The general goal of reinforcement learning is to optimize the expected return of policy π, denoted as $\mathbb{E}[J(\pi)]$. Many works utilize a Monte-Carlo estimate of the policy gradient

$$\nabla_\phi \mathbb{E}\Big[J(\pi_\phi)\Big] = \mathbb{E}\Big[\nabla_\phi \sum_{(\mathrm{s},\mathrm{a})\in\tau} \log \pi_\phi(\mathrm{a}|\mathrm{s}) r(\mathrm{s},\mathrm{a})\Big]. \tag{3}$$

The expectation is taken over the trajectory distribution $\eta^\pi(\tau)$, i.e. distribution of sequences of (s, a). In the context of ToD, we may cast the state s as being the underlying dialogue state which may be unobserved, in which case the problem becomes partially observable. In ToD systems based on word-level RL an action a corresponds to predicting the next word out of a large vocabulary, where in latent-action RL a is cast to predicting one element of a latent-space representation. In our setting, the latent dialogue policy is warm-started by the parameters of the variational posterior from the SL stage.

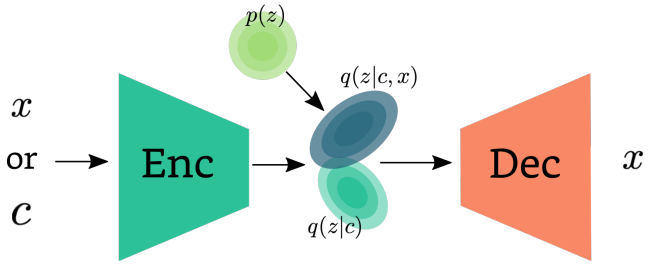


Figure 1: *SL Stage* Schema. The encoder receives either context or both context and response that is to be decoded from latent z. When samples from $q(\mathrm{z}|\mathrm{c},\mathrm{x})$ are used for decoding, the architecture acts as a proper conditional autoencoder.

3 Method

Our method, called TCUP, is in line with the general latent action RL paradigm outlined in Sec. 2, i.e. TCUP is also based on a two-stage approach. However, we propose contributions to both stages, which significantly extend prior work. In Sec. 3.1 we describe a reformulation of the ELBO for the *SL Stage*, which is vital to reach state-of-the-art performance with continuous variational posteriors. In Sec. 3.2 we discuss our reinforcement learning setup and introduce methods to address the deterioration of response coherence in *latent action*-based dialogue policies through regularization.

3.1 *SL Stage* – Revisiting the Full ELBO

As aforementioned, prior work by Zhao, Xie, and Eskenazi (2019) introduced the "lite" ELBO (described in equation 2), to alleviate the overexposure bias that emerges when response information is incorporated in the optimization objective. Indeed, if we only optimize for samples of $q(\mathrm{z}|\mathrm{c},\mathrm{x})$, the most information about the responses x is found within the responses themselves, which could encourage the model to ignore the context c. However, not conditioning on x will reduce the expressiveness of the variational posterior for capturing generative factors of responses and for predicting the best next response given contexts. In this work we rely on a full ELBO and introduce a conditional prior $q_\zeta^p(\mathrm{z}|\mathrm{c})$ indicated by the superscript p which is constrained by the free prior $p(\mathrm{z})$ and also parametrized.

Proposition 1. *For the minimization problem* $\min_q D_{\mathrm{KL}}[q||p(\mathrm{z}|\mathrm{x},\mathrm{c})]$*, given sufficiently similar* $p(z|c)$ *and* $p(z|x)$*, it suffices to minimize*

$$\begin{aligned}D_{\mathrm{KL}}[q_\theta||q_\zeta^p] + D_{\mathrm{KL}}[q_\zeta^p||p(\mathrm{z})] - \\ \mathbb{E}_{q_\theta}[\log p_\phi(\mathrm{x}|\mathrm{z})] - \mathbb{E}_{q_\zeta^p}[\log p_\phi(\mathrm{x}|\mathrm{z})] + C,\end{aligned} \tag{4}$$

with respect to q_θ *and* q_ζ^p.

Prop. 1 allows us to optimize equation 4 in the *SL Stage*, we defer its proof to the Appendix Sec. A. While equation 4 is agnostic to the choice of $p(\mathrm{z})$, we base our approach on a multivariate, isotropic Gaussian distribution. The benefits of incorporating response information is also confirmed by prior work LAVA (Lubis et al. 2020). However, to make sure the decoder doesn't overly rely on information from x, LAVA needs to introduce two auxiliary tasks, i.e. one auto-encoding the responses and one for generating responses based on context information. We achieve this by leveraging samples from the prior and the variational posterior for reconstruction during training time. This considerably simplifies the training procedure and improves performance (Sec. 4).

3.2 *RL Stage* – Regularizing for Success and Coherence

Dialogue success, i.e. the accomplishment of a user's task, is the most important reward signal for training RL polices for ToD systems. The success signal is defined as a system's ability to fill all pre-defined reference slots that are necessary to fulfill the user's goal, with the correct values. In a restaurant booking task the slots can be food type, reservation time, etc. These values are observed at the end of the dialogue and are usually inferred from the generated responses (Budzianowski et al. 2018). Success rate is the fraction of dialogues in which the task was successfully accomplished.

By checking for relevant slots to be present in the response, success rate permits that long, possibly incoherent responses achieve success more easily due to higher chance of containing the correct slot tokens. For measuring response coherence, we rely on the BLEU metric as specified in the MultiWOZ benchmark (Budzianowski et al. 2018). Anecdotal evidence for this phenomenon is presented in Tab. **??**. This is not uncommon in reinforcement learning algorithms, which have shown to exploit weaknesses in simulators of game environments (Mnih et al. 2013) by finding high reward states which don't align with solving the underlying task. In the context of latent action RL for ToD, the decoder can be seen as a weak simulator that is prone to exploitation due to the nature of the success rate metric. In the following subsections we describe three counter measures to address this challenge.

Penalizing Out of Distribution Samples Depending on the choice of distribution for the latent actions (e.g. gaussians), the aforementioned issue can be even more severe, since the model is able to sample out of distribution (OOD) utterances that provide success. In the *SL Stage* we apply variational inference, which implicitly maximizes the BLEU score through maximizing the ELBO (see equation 4). We denote this BLEU-maximizing policy with π^{VI}. Since we constrain our policy with an isotropic Gaussian prior in the first training stage, we can leverage this information to prevent the policy from deviating from this prior in the form of a divergence cost which can be efficiently computed. Concretely, the regularized reward function is defined as

$$r(\mathrm{x}, \mathrm{c}) = succ(\mathrm{x}) - \beta D_{\mathrm{KL}}[\pi(\mathrm{z}|\mathrm{c}) \,||\, p(\mathrm{z})], \tag{5}$$

where $succ(\mathrm{x})$ is an indicator function

$$succ(\mathrm{x}) = \begin{cases} 1 \text{ response x entails correct reference slots} \\ 0 \text{ otherwise.} \end{cases} \tag{6}$$

ToD as Offline RL We are dealing with an offline reinforcement learning problem, since we have a dataset of optimal responses without the possiblity of obtaining more samples via a simulator or users. By shifting the reinforcement learning problem to the latent space, we are implicitly creating a surrogate online problem, where we need to obtain samples form $\pi(\mathrm{z}|\mathrm{c})$ and evaluate them.

We argue that one of the reasons why Gausssian latent spaces have been reported as under-performing in comparison to categoricals is the biased and noisy gradient estimate based on samples from a single dialogue. Contrary to prior work (Lubis et al. 2020; Zhao, Xie, and Eskenazi 2019), which estimates the policy gradient over the responses from a single dialogue sample, we take advantage of a Monte Carlo policy gradient estimate across *multiple dialogues*.

Replaying Succesful Samples Re-using encountered experience by storing it in memory (a replay buffer) has proved to be beneficial for sample-efficiency in reinforcement learning. However, a naive usage of the buffer has multiple caveats in the MultiWOZ setting. Firstly, since the success signal is calculated on the dialogue level, some responses, that might be successful conditioned on the dialogue state and context, might be labeled as negative. Intuitively, the policy can start off the dialogue correctly, but fail to complete it. This can lead to conflicting examples in the buffer, which destabilizes training. Secondly, we have a many-to-one mapping from responses to success, which leads to multimodality, but also modes that might be incoherent. If the response at time step t is conditionally independent of the dialogue history given the dialogue state and input utterance, we can attribute success directly to the response independent of past utterances. This motivates the storage of only successful responses in the replay buffer, which we sample as a fraction of the training batch. It mitigates the problem of false negative responses and ensures fewer conflicting examples in the batch. Replaying past experience ensures that certain samples are not forgotten, which increases training stability and ensures that the current policy π stays close to π^{VI}. In practice, we exchange a certain sample in the batch generated by the current policy with a sample from the replay buffer with probability λ. As $\lambda \to 0$ we arrive at the simple REINFORCE update, $\lambda = 1$ means that we only use replayed samples for updates.

4 Experiments

We provide a detailed evaluation of TCUP's dialogue policy in Sec. 4.1. This includes comparing its performance on the MultiWOZ benchmark; an ablation study to assess the importance of our technical contributions from Sec. 3; and a qualitative analysis of response coherence. Finally, we analyze the quality of the latent representations in Sec. 4.2 and provide evidence that TCUP learns representations, which yield good separation and clustering of MultiWOZ domains and actions. We use a recurrent encoder-decoder architecture

Model	BLEU	Inform	Success	Av. len.	CBE	# unigrams	# trigrams
MarCo*	17.3	94.5	87.2	16.01	1.94	319	3002
HDSA	**20.7**	87.9	79.4	14.42	1.64	259	2019
HDNO†	17.8	93.3	83.4	14.96	0.84	103	315
SFN†	14.1	93.4	82.3	14.93	1.63	188	1218
UniConv	18.1	66.7	58.7	14.17	1.79	338	2932
LAVA$^{\dagger *}$	16.2	89.7	77.6	14.41	1.96	272	2365
LAVA†-Cat	10.8	95.9	93.5	13.28	1.27	176	708
TCUP†	10.3	**96.3**	**95.9**	15.14	1.44	210	1838
TCUP†-Cat	14.3	96.1	92.1	14.33	1.61	230	1490

Table 1: Competitor comparison on MultiWOZ. RL methods are marked with †, transformer architectures with $\star$.

USER	we are staying [value_count] people for [value_count] nights starting from [value_day] . i need the reference number?		
NAIVE SYSTEM	[hotel_name] [hotel_name] [hotel_name] [hotel_reference] [hotel_name] [hotel_reference] [hotel_name] [hotel_reference] [hotel_name] [hotel_phone] . . .	TCUP	[hotel_name] is located at [hotel_address]. their phone number is [hotel_phone]. is there anything else i can help with?
USER	just the address please		
NAIVE SYSTEM	[attraction_name] is located at [attraction_address], postcode [attraction_postcode]. the phone number is [attraction_phone]. [attraction_name] is located . . .	TCUP	[attraction_name] s phone number is [attraction_phone]. it s located on [attraction_address], and their phone number is [attraction_phone].

Table 2: Example of successful responses, which are generated by a naive system policy "gaming" the metric and producing incoherent utterances with low BLEU scores together with the corresponding TCUP responses.

with dot-product attention, further details on the architecture and training procedure can be found in Appendix Sec. D.

SL Stage Here, we learn the mapping from context to response using equation 1. In practice, we optimize the $p_\phi(\mathrm{x}|\mathrm{z})$ part with a weighted cross-entropy loss that puts higher weights on slot placeholders in the response. The best model is selected based on its BLEU score.

RL Stage In this stage we fix the parameters of the decoder and train only the encoder parameters via policy gradient with two important modifications: (i) we use a batched version of the policy gradient which contains samples from multiple dialogues and hence reduces the variance of the gradient, and (ii) in each batch we sample a mix of newly generated and old experience (with which we have obtained a success signal earlier). This implicitly keeps us close to the starting policy which results in more stable training. It is beneficial to replace the standard $D_{\mathrm{KL}}[q||p]$ term of the variational objective with a symmetric version $\frac{1}{2}(D_{\mathrm{KL}}[q||p] + D_{\mathrm{KL}}[p||q])$. This ensures that regions where the densities of p and q behave differently are treated equally irrespective of the ordering.

4.1 Context-to-Response Generation

We evaluate TCUP using MultiWOZ 2.1 (Wang et al. 2020) on the policy learning task for context to response generation. MultiWOZ contains 10438 dialogues across six different domains, pre-split into 8438 training, 1000 validation, and 1000 testing records. The task's objective is to generate the next response at every system turn given the ground-truth dialog belief state and the previous dialogue. For a fair comparison we use the same delexicalization approach as in prior work (Lubis et al. (2020)). Additionally to our proposed approach based on isotropic Gaussians, we introduce a variant which is based on categorical latent distributions (coined TCUP-Categorical). Table 1 shows that TCUP improves the state-of-the-art inform- and success rate metric across all competitors. Moreover, it is also competitive in terms of language diversity metrics. Compared to the currently best performing latent action reinforcement learning approach (LAVA), we increase all metrics except for minor decreases in the BLEU metric. The response coherence is further discussed in Sec. 4.1.

Latent Representations While prior latent-action RL approaches(Lubis et al. 2020; Zhao, Xie, and Eskenazi 2019) favor categorical latent distributions with modified attention mechanisms in the decoder, our results demonstrate that relying on isotropic Gaussian latents is advantageous, even if we only use simple dot-product attention in the decoder. Focusing on TCUP-Categorical we observe competitive results in terms of inform and success rate compared to prior work, but inferior performance compared to TCUP with continuous latents. Using categoricals limits us in the sense that we don't make full use of the nonlinearity in the decoder to decode diverse responses, which manifests itself with poor results in the diversity metrics (Tab. 1). By using a continuous distribution, we improve the diversity of the generated responses

*Results taken for best runs, mean performance is actually lower, additional commentary available in Appendix Sec. B

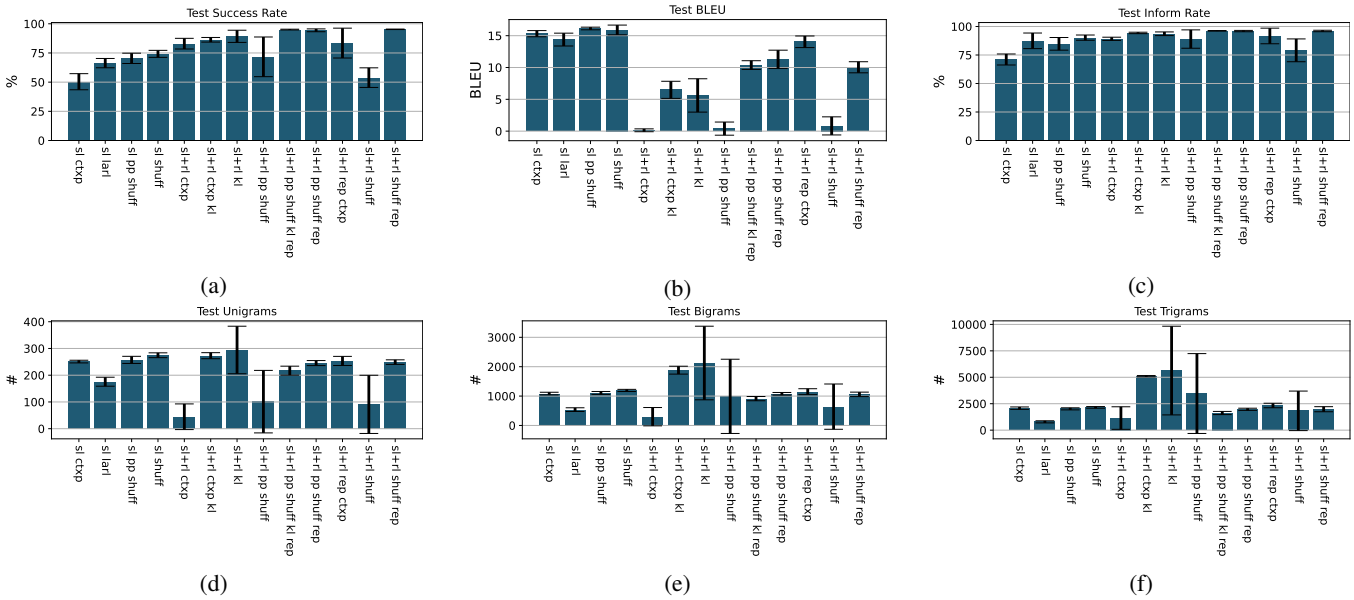


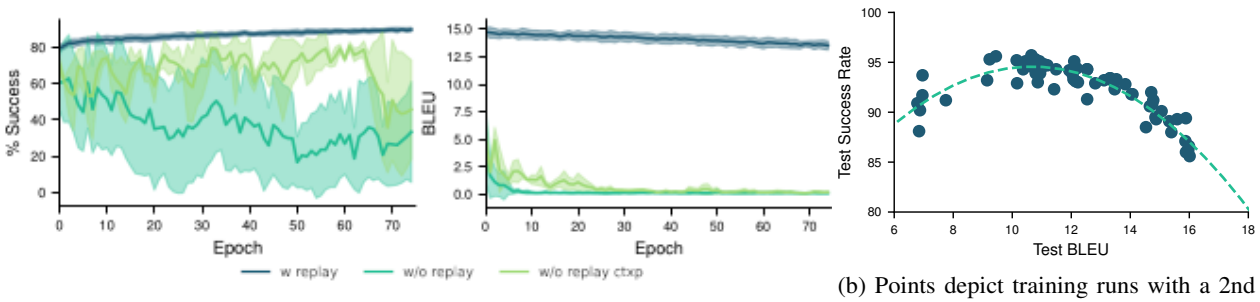
Figure 2: Ablation study. TCUP configurations are averaged over three random seeds. *sl* denotes supervised training, *sl+rl* supervised stage followed by reinforcement learning *sl+larl* work in (Zhao, Xie, and Eskenazi 2019), *ctxp* contextual prior without prior on it (q^p_ζ, equation 4), *pp* identity covariance Gaussian prior on contextual prior ($D_{\mathrm{KL}}[q^p_\zeta||p(\mathrm{z})]$, equation 4), *shuff* prior and posterior shuffling (Sec. 3.1), *rep* replay buffer (Sec. 3.2), *kl* penalty term to identity covariance Gaussian (Sec. 3.2)

compared to those two competitors.

Ablation Study Fig. 2 presents an ablation study over different variants of TCUP. Variants only using the *SL Stage* (*sl*) have the highest BLEU metrics, but show low success rates compared to approaches with a *RL Stage* (*sl + rl*). Our experiments demonstrate that the proposed variational inference objective Sec. 3.1, i.e. the full ELBO with prior and posterior shuffling (*sl shuff*) and an identity covariance Gaussian prior on the contextual prior (*sl pp shuff*) leads to improvements across all metrics compared to the objective used in prior work (*sl larl*) LaRL (Zhao, Xie, and Eskenazi 2019) and LAVA (Lubis et al. 2020). Among the two, *sl shuff* performs better in the success and inform metrics, where *sl pp shuff* delivers minor improvements in BLEU. Through adding the *RL Stage* (*sl + rl*), significantly higher success rates are achieved while sacrificing BLEU. The BLEU decrease is alleviated by applying optimal replay (all variants with *rep* in their name) and the constrained rewards with the KL penalty (*kl*), with replay being the superior choice between the two. For instance, compare the scores of *sl + rl ctxp rep* and *sl + rl ctxp kl*. Further gains can be achieved by utilizing both *sl + rl pp shuff kl rep*. Not using optimal replay introduces higher variance across runs. Considering the variants with highest bi- and trigrams (*sl + rl kl*, *sl + rl ctxp kl*) we observe that high scores in these metrics are not indicative of high coherence, if we compare with their corresponding BLEU scores. In summary, the results (*shuff kl rep*) validate the benefits of our variational inference objective and regularization.

Response Coherence We observed that it is possible to maximize success rates at the cost of lower coherence in terms of BLEU scores. Fig. 3a shows that we achieve state-of-the-art success rates, whilst achieving a BLEU score of effectively 0. This is an artifact of the success rate metric, i.e. it disregards response coherence and only checks if the correct slot values have been addressed by the dialogue policy.

In Fig. 3b we see different runs of our method with different strength of regularization in terms of KL penalty and optimal replay fraction. Depending on the strength of regularization they form a Pareto front. This further shows the multi-objective trade-off between success rates and BLEU scores. The BLEU collapse is to be expected when using latent-action reinforcement learning, since longer and more diverse answers have positive impact on the dialogue success, which leads to the policy selecting degenerate responses. Furthermore, if no additional regularization is used, the latent policy learns to sample outliers in terms of the prior over z. We argue that it is not realistic to expect the policy resulting from the reinforcement learning stage to outperform the response coherence of the supervised learning stage in terms of BLEU (as long as the primary purpose of the RL stage is to improve dialogue success metrics). Instead, we show that through regularization and optimal replay, the BLEU score can be kept from deterioration through the RL stage while we optimize dialogue level metrics. An alternative approach to alleviating degenerate policies would be to simply make the BLEU score or other coherence metrics part of the reward function. For example, the final comparison of models in (Budzianowski et al. 2018) is done by comparing $\frac{success+inform}{2} + BLEU$. However, we suggest that there are multiple problems with the approach of maximizing such a hybrid metric directly. First of all, constructing reliable metrics is challenging (Jiang et al. 2021; Mehri and Eskenazi



(a) Influence of replay on success rate and BLEU

(b) Points depict training runs with a 2nd degree polynomial showing the Pareto front.

Figure 3: Interdependencies between success rate and BLEU scores

2020) in itself. More importantly, as soon as the coherence metric is part of the reward, we encounter the problem of adequate scaling in comparison to the success rate.

Impact of Regularization When introducing regularization in the form of optimal replay sampling and the KL penalty term, we are able to achieve state-of-the-art performance in terms of success rate without lowering the BLEU score significantly. As depicted in Fig. 3a our method using optimal replay buffer sampling is more stable during training in comparison to the naive application of the policy gradient. Also, we observe that the BLEU deterioration is kept at bay and roughly deteriorates linearly over time. By increasing the replay fraction λ too much we are over-constraining the latent dialogue policy to the initial experience, which leads to biased updates and hurts exploration. Nevertheless, as shown in the ablation study (Fig. 2) optimal replay and the KL penalty term are essential for preventing the BLEU score from deteriorating too rapidly. In this work we aimed to retain a BLEU score that is competitive to that reported by LAVA (Lubis et al. 2020), while improving overall dialogue success. A sensitivity analysis over the weights of the penalty term and the replay fraction λ is described in Appendix Fig. 6.

4.2 Latent Space Analysis

Fig. 4 depicts a UMAP (McInnes, Healy, and Melville 2018) projection of the learned latent samples z for the Gaussian case. We observe similar behavior to Lubis et al. (2020). In the supervised learning stage there is apparent clustering in terms of domain labels (Fig. 4a). The reinforcement learning stage with regularization leads to specialization of the clusters (Fig. 4c). In comparison, a good cluster separation is lost (Fig. 4b) without applying regularization in the RL stage. This can be explained by the fact that the z samples are degenerate samples that lie in low-support regions of $q_\zeta^p(\mathrm{z}|\mathrm{c})$.

We have calculated the Caliński-Harabasz index (Caliński and Harabasz 1974), otherwise known as Variance Ratio Criterion, to evaluate the clustering in the latent space with respect to domain and action type labels taken from DAMD (Zhang, Ou, and Yu 2020). In comparison to the results reported by Lubis et al. (2020), our model is able to obtain high scores in the supervised stage of training already. The scores drop slightly after RL fine-tuning, but remain at a higher level than those for categorical latents. It's important to note that the scores for LaRL* and LAVA* were computed with such categoricals, whereas our scores are based on Gaussian latents, which are continuous and unbounded. Tab. 3 demonstrates the value in using continuous latent representations, rather than to make direct numeric comparisons.

Table 3: Caliński-Harabasz scores (higher is better).

Model	SL		RL	
	Domain	Action	Domain	Action
LaRL*	93.19	23.20	121.15	17.5
LAVA*	104.92	25.28	158.00	41.75
TCUP	345.48	80.76	329.39	78.95

5 Related Work

Supervised Learning The majority of prior work applies some form of Supervised Learning. For an extensive overview we refer to Gao, Galley, and Li (2018) and focus on the state-of-the-art competitors HDSA (Chen et al. 2019) as well as UniConv (Le et al. 2020). Both approaches demonstrate the benefits of jointly training multiple dialogue tasks at once, such as predicting dialogue acts and states.

Variational Inference Inspired by variational autoencoders (VAEs) (Kingma and Welling 2013), several works employ variational inference for learning conditional response distributions. VHRED (Serban et al. 2017) is a variational hierarchical RNN for modelling dependencies between words and utterances. Stochastic latent variables are introduced to generate the next utterance. To avoid collapsing posteriors Zhao, Zhao, and Eskenazi (2017) learn response representations using an auxiliary task introducing a bag of words loss. Shen et al. (2018) improves the stability of VHRED by splitting the training process into two parts. First, text is auto-encoded into continuous embeddings, which are the starting point for learning latent representations by reconstructing the embeddings. To address the degeneration problem, Variational Hierarchical Conversation RNNs (VHCR) (Park, Cho, and Kim 2018) exploits an utterance drop regularization. DialogWAE (Gu et al. 2018) represents the prior distribution as Gaussian mixture and adapts WGAN (Arjovsky, Chintala,

*Best scores of each method obtained by Lubis et al. (2020)

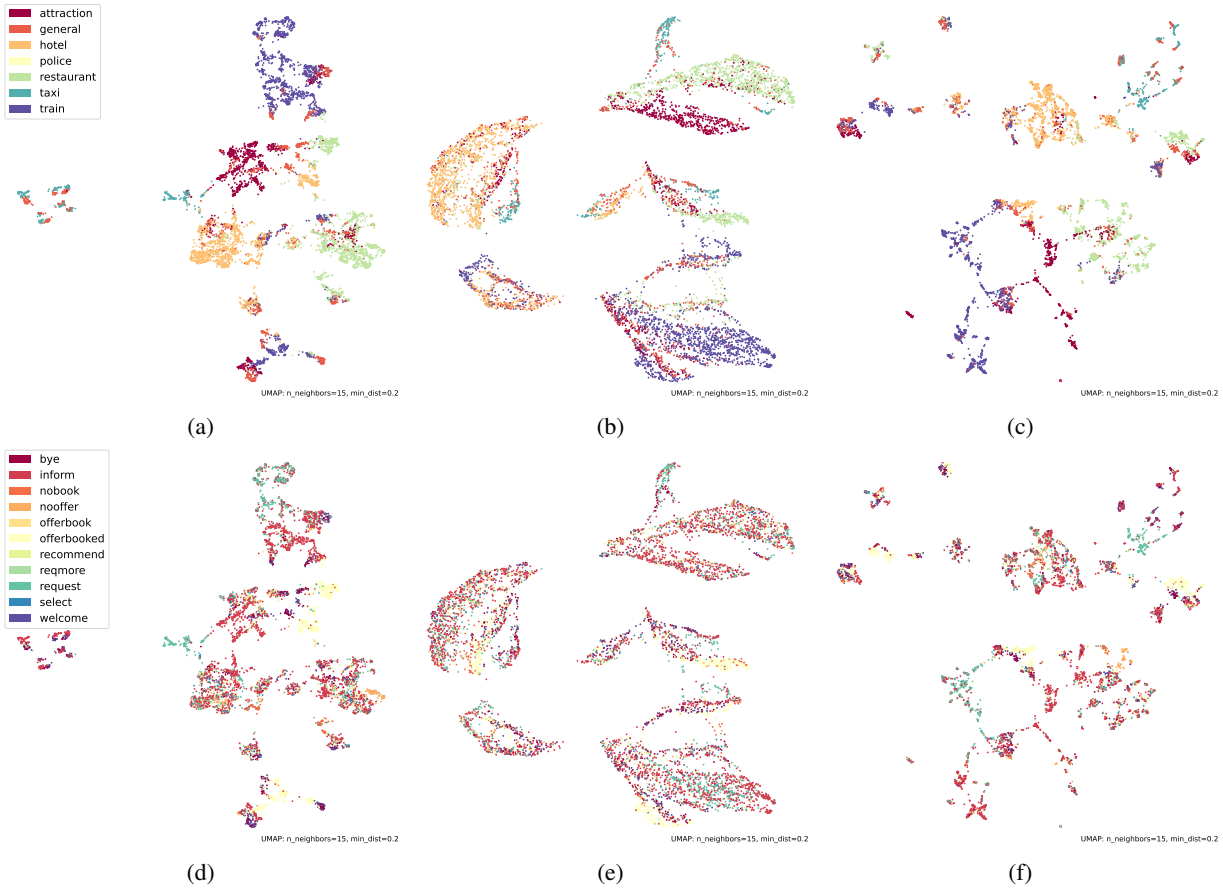


Figure 4: UMAP embeddings of latent representations. Figures a-c: domain labeled embeddings for the SL, SL+RL and SL+RL with replay sampling, bottom row: same embeddings labeled by response type. Executing the RL stage of the training results in representations that are difficult to separate (b and e). By applying replay we obtain higher specialization in the clusters (c and f).

and Bottou 2017) for training.

Combined Supervised- & Reinforcement Learning Henderson, Lemon, and Georgila (2008) was the first work to combine Reinforcement and Supervised Learning introducing a value function, which relies on SL for predicting the expected future reward of states not covered by the data directly. Williams, Asadi, and Zweig (2017) prevent high variance by performing a SL gradient update, if the RL policy output deviates from the training data. Fatemi et al. (2016); Su et al. (2017) apply two-stage approaches, where they pretrain a dialogue policy supervised, which is then further optimized by RL. HappyBot (Shin et al. 2019) relies on a weighted combination of a maximum likelihood and a REINFORCE objective. Saleh et al. (2020) uses REINFORCE-based policy gradients to update the prior probability distribution of the latent variational model trained using supervised learning. Structured Fusion Networks (SFNs) (Mehri, Srinivasan, and Eskenazi 2019) apply RL to fuse dialogue modules, where each module serves a different purpose (NLU, NLG, etc.) and is pretrained in individual supervised stages. Apart from task-oriented dialogues a combination of SL and RL has also been applied for generating responses for open-domain dialogues (Xu, Wu, and Wu 2018). The aforementioned approaches are based on word-level RL, suffering from huge action-spaces covering the entire input vocabulary. Due to this, ensuring coherent responses is challenging (Lewis et al. 2017; Kottur et al. 2017), especially in multi-turn dialogues spanning hundreds of words. LaRL (Zhao, Xie, and Eskenazi 2019) and LAVA (Lubis et al. 2020) address this problem by learning a low-dimensional latent representation with amortized variational inference followed by RL fine-tuning.

6 Conclusion

We showed that with appropriate modifications to the reinforcement learning procedure and the latent space structure, it is possible to obtain state-of-the-art results with simple Gaussian latent distributions. The problem of coherence deterioration when optimizing for success rate points to the fact that better metrics are needed for developing efficient dialogue policies - we were able to all eviate this with optimal replay and KL regularization. Although TCUP shows promising results in the policy learning setting with access to ground-truth dialogue state, it would be interesting to see how TCUP compares in the end-to-end learning setting.

References

Arjovsky, M.; Chintala, S.; and Bottou, L. 2017. Wasserstein generative adversarial networks. In *International conference on machine learning*, 214–223. PMLR.

Budzianowski, P.; Wen, T.-H.; Tseng, B.-H.; Casanueva, I.; Ultes, S.; Ramadan, O.; and Gašić, M. 2018. MultiWOZ– A Large-Scale Multi-Domain Wizard-of-Oz Dataset for Task-Oriented Dialogue Modelling. *arXiv preprint arXiv:1810.00278*.

Caliński, T.; and Harabasz, J. 1974. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1): 1–27.

Chen, W.; Chen, J.; Qin, P.; Yan, X.; and Wang, W. Y. 2019. Semantically conditioned dialog response generation via hierarchical disentangled self-attention. *arXiv preprint arXiv:1905.12866*.

Cho, K.; van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; and Bengio, Y. 2014. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1724–1734.

Fatemi, M.; Asri, L. E.; Schulz, H.; He, J.; and Suleman, K. 2016. Policy networks with two-stage training for dialogue systems. *arXiv preprint arXiv:1606.03152*.

Gao, J.; Galley, M.; and Li, L. 2018. Neural approaches to conversational ai. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, 1371–1374.

Gao, J.; Galley, M.; and Li, L. 2019. *Neural Approaches to Conversational AI: Question Answering, Task-oriented Dialogues and Social Chatbots*. now.

Gu, X.; Cho, K.; Ha, J.-W.; and Kim, S. 2018. Dialogwae: Multimodal response generation with conditional wasserstein auto-encoder. *arXiv preprint arXiv:1805.12352*.

Henderson, J.; Lemon, O.; and Georgila, K. 2008. Hybrid reinforcement/supervised learning of dialogue policies from fixed data sets. *Computational Linguistics*, 34(4): 487–511.

Henderson, M.; Vulić, I.; Gerz, D.; Casanueva, I.; Budzianowski, P.; Coope, S.; Spithourakis, G.; Wen, T.-H.; Mrkšić, N.; and Su, P.-H. 2019. Training neural response selection for task-oriented dialogue systems. *arXiv preprint arXiv:1906.01543*.

Hochreiter, S.; and Schmidhuber, J. 1997. Long short-term memory. *Neural computation*, 9(8): 1735–1780.

Inaba, M.; and Takahashi, K. 2016. Neural utterance ranking model for conversational dialogue systems. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, 393–403.

Jiang, H.; Dai, B.; Yang, M.; Zhao, T.; and Wei, W. 2021. Towards Automatic Evaluation of Dialog Systems: A Model-Free Off-Policy Evaluation Approach. *arXiv preprint arXiv:2102.10242*.

Kingma, D. P.; and Welling, M. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Kottur, S.; Moura, J. M.; Lee, S.; and Batra, D. 2017. Natural language does not emerge'naturally'in multi-agent dialog. *arXiv preprint arXiv:1706.08502*.

Le, H.; Sahoo, D.; Liu, C.; Chen, N. F.; and Hoi, S. C. 2020. Uniconv: A unified conversational neural architecture for multi-domain task-oriented dialogues. *arXiv preprint arXiv:2004.14307*.

Lee, H.; Jo, S.; Kim, H.; Jung, S.; and Kim, T.-Y. 2021. SUMBT+ LaRL: Effective Multi-Domain End-to-End Neural Task-Oriented Dialog System. *IEEE Access*, 9: 116133–116146.

Lewis, M.; Yarats, D.; Dauphin, Y. N.; Parikh, D.; and Batra, D. 2017. Deal or no deal? end-to-end learning for negotiation dialogues. *arXiv preprint arXiv:1706.05125*.

LI, S.; Yavuz, S.; Hashimoto, K.; Li, J.; Niu, T.; Rajani, N.; Yan, X.; Zhou, Y.; and Xiong, C. 2020. CoCo: Controllable Counterfactuals for Evaluating Dialogue State Trackers. In *International Conference on Learning Representations*.

Lubis, N.; Geishauser, C.; Heck, M.; Lin, H.; Moresi, M.; van Niekerk, C.; and Gasic, M. 2020. LAVA: Latent Action Spaces via Variational Auto-encoding for Dialogue Policy Optimization. In Scott, D.; Bel, N.; and Zong, C., eds., *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, 465–479. International Committee on Computational Linguistics.

McInnes, L.; Healy, J.; and Melville, J. 2018. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*.

Mehri, S.; and Eskenazi, M. 2020. Unsupervised evaluation of interactive dialog with dialogpt. *arXiv preprint arXiv:2006.12719*.

Mehri, S.; Srinivasan, T.; and Eskenazi, M. 2019. Structured fusion networks for dialog. *arXiv preprint arXiv:1907.10016*.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; and Riedmiller, M. 2013. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.

Park, Y.; Cho, J.; and Kim, G. 2018. A hierarchical latent structure for variational conversation modeling. *arXiv preprint arXiv:1804.03424*.

Saleh, A.; Jaques, N.; Ghandeharioun, A.; Shen, J.; and Picard, R. 2020. Hierarchical reinforcement learning for open-domain dialog. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 8741–8748.

Serban, I.; Sordoni, A.; Lowe, R.; Charlin, L.; Pineau, J.; Courville, A.; and Bengio, Y. 2017. A hierarchical latent variable encoder-decoder model for generating dialogues. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

Shen, X.; Su, H.; Niu, S.; and Demberg, V. 2018. Improving variational encoder-decoders in dialogue generation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Shin, J.; Xu, P.; Madotto, A.; and Fung, P. 2019. Happybot: Generating empathetic dialogue responses by improving user experience look-ahead. *arXiv preprint arXiv:1906.08487*.

Su, P.-H.; Budzianowski, P.; Ultes, S.; Gasic, M.; and Young, S. 2017. Sample-efficient actor-critic reinforcement learning with supervised data for dialogue management. *arXiv preprint arXiv:1707.00130*.

Tao, C.; Wu, W.; Xu, C.; Hu, W.; Zhao, D.; and Yan, R. 2019. Multi-representation fusion network for multi-turn response selection in retrieval-based chatbots. In *Proceedings of the twelfth ACM international conference on web search and data mining*, 267–275.

Walker, M. A.; Stent, A.; Mairesse, F.; and Prasad, R. 2007. Individual and domain adaptation in sentence planning for dialogue. *Journal of Artificial Intelligence Research*, 30: 413–456.

Wang, K.; Tian, J.; Wang, R.; Quan, X.; and Yu, J. 2020. Multi-domain dialogue acts and response co-generation. *arXiv preprint arXiv:2004.12363*.

Williams, J. D.; Asadi, K.; and Zweig, G. 2017. Hybrid code networks: practical and efficient end-to-end dialog control with supervised and reinforcement learning. *arXiv preprint arXiv:1702.03274*.

Xu, C.; Wu, W.; and Wu, Y. 2018. Towards explainable and controllable open domain dialogue generation with dialogue acts. *arXiv preprint arXiv:1807.07255*.

Yan, R.; Song, Y.; and Wu, H. 2016. Learning to respond with deep neural networks for retrieval-based human-computer conversation system. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, 55–64.

Zhang, Y.; Ou, Z.; and Yu, Z. 2020. Task-oriented dialog systems that consider multiple appropriate responses under the same context. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 9604–9611.

Zhao, T.; Xie, K.; and Eskenazi, M. 2019. Rethinking Action Spaces for Reinforcement Learning in End-to-end Dialog Agents with Latent Variable Models. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 1208–1218. Minneapolis, Minnesota: Association for Computational Linguistics.

Zhao, T.; Zhao, R.; and Eskenazi, M. 2017. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. *arXiv preprint arXiv:1703.10960*.

A Revisiting the Full ELBO

Here we provide proof of proposition 1. Further we use shorthand q for $q(\mathrm{z}|\mathrm{x},\mathrm{c})$. Firstly we define our optimization problem

$$\min_q D_{\mathrm{KL}}[q||p(\mathrm{z}|\mathrm{x},\mathrm{c})].$$

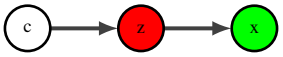


Figure 5: Probabilistic model of dependencies between c, z and x

Lemma 1. *Given the probabilistic model diagram 5:*

$$D_{\mathrm{KL}}[q||p(\mathrm{z}|\mathrm{x},\mathrm{c})] \leq \mathbb{E}_q[\log q] - \mathbb{E}_q[\log p(\mathrm{x}|\mathrm{z})] - \mathbb{E}_q[\log p(\mathrm{z}|\mathrm{c})]$$

Proof. We expand the KL divergence of equation 7.

$$D_{\mathrm{KL}}[q||p(\mathrm{z}|\mathrm{x},\mathrm{c})] = \mathbb{E}_q[\log q - \log p(\mathrm{z}|\mathrm{x},\mathrm{c})] \tag{7}$$

We first expand the $p(\mathrm{z}|\mathrm{x},\mathrm{c})$ by Bayes rule and make use of independence between x and c,

$$\begin{aligned} p(\mathrm{z}|\mathrm{x},\mathrm{c}) &= \frac{p(\mathrm{x},\mathrm{c}|\mathrm{z})p(\mathrm{z})}{p(\mathrm{x},\mathrm{c})} \\ &= \frac{p(\mathrm{x}|\mathrm{z})p(\mathrm{c}|\mathrm{z})p(\mathrm{z})}{p(\mathrm{x},\mathrm{c})},\ \mathrm{x} \perp\!\!\!\perp \mathrm{c}|\,\mathrm{z} \\ &= \frac{p(\mathrm{x}|\mathrm{z})p(\mathrm{z}|\mathrm{c})p(\mathrm{c})}{p(\mathrm{x},\mathrm{c})}. \end{aligned}$$

Putting back these terms into equation 7 leads to

$$\begin{aligned} &D_{\mathrm{KL}}[q||p(\mathrm{z}|\mathrm{x},\mathrm{c})] = \\ &\mathbb{E}_q[\log q] - \mathbb{E}_q[\log p(\mathrm{x}|\mathrm{z})] - \mathbb{E}_q[\log p(\mathrm{z}|\mathrm{c})] \\ &- \mathbb{E}_q[\log p(\mathrm{c})] + \mathbb{E}[\log p(\mathrm{x},\mathrm{c})] \\ &= \mathbb{E}_q[\log q] - \mathbb{E}_q[\log p(\mathrm{x}|\mathrm{z})] - \mathbb{E}_q[\log p(\mathrm{z}|\mathrm{c})] \\ &- \log p(\mathrm{c}) + \log p(\mathrm{x},\mathrm{c}) \\ &\leq \mathbb{E}_q[\log q] - \mathbb{E}_q[\log p(\mathrm{x}|\mathrm{z})] \\ &- \mathbb{E}_q[\log p(\mathrm{z}|\mathrm{c})]\, p(\mathrm{c}) \geq p(\mathrm{x},\mathrm{c}). \end{aligned}$$

□

In practice, we don't have access to $p(\mathrm{z}|\mathrm{c})$ and hence we estimate it with $q_\theta^p(\mathrm{z}|\mathrm{c})$, which results in an $\arg\min$ operation within the expectation.

$$\begin{aligned} &\mathbb{E}_q[\log q] - \mathbb{E}_q[\log p(\mathrm{x}|\mathrm{z})] - \mathbb{E}_q[\log p(\mathrm{z}|\mathrm{c})] = \\ &\mathbb{E}_q[\log q] - \mathbb{E}_q[\log \arg\min_{q^p} D_{\mathrm{KL}}[q^p||p(\mathrm{z}|\mathrm{c})]] \\ &- \mathbb{E}_q[\log p(\mathrm{x}|\mathrm{z})] \end{aligned}$$

The new KL term within the expectation can be expanded to obtain the traditional evidence lower-bound, ie. bound on the objective:

$$\begin{aligned} D_{\mathrm{KL}}[q^p||p(\mathrm{z}|\mathrm{c})] \leq & D_{\mathrm{KL}}[q_\zeta^p||p(\mathrm{z})] \\ &+ \mathbb{E}_{q_\zeta^p}[p(\mathrm{c}|\mathrm{z})] \end{aligned}$$

In the end, this is our objective function, but we want to get rid of $p_\phi(\mathrm{c}|\mathrm{z})$.

$$\begin{aligned} \mathcal{L}(\phi,\theta,\zeta) = & D_{\mathrm{KL}}[q_\theta||q_\zeta^p] + D_{\mathrm{KL}}[q_\zeta^p||p(\mathrm{z})] - \\ & \mathbb{E}_{q_\theta}[\log p_\phi(\mathrm{x}|\mathrm{z})] - \mathbb{E}_{q_\zeta^p}[\log p(\mathrm{c}|\mathrm{z})] \end{aligned}$$

The current formulation would require us to approximate $p(\mathrm{c}|\mathrm{z})$.

Remark. *The KL term $D_{\mathrm{KL}}[p(\mathrm{z}|\mathrm{c})||p(\mathrm{z}|\mathrm{x},\mathrm{c})]$ is zero iff* x *doesn't provide any additional information about* z*.*

We further simplify the problem by replacing $p(\mathrm{c}|\mathrm{z})$ with $p_\phi(\mathrm{x}|\mathrm{z})$. By applying Bayes rule, we arrive to the following:

$$\begin{aligned} \mathcal{L}(\phi,\theta,\zeta) = & D_{\mathrm{KL}}[q_\theta||q_\zeta^p] + D_{\mathrm{KL}}[q_\zeta^p||p(\mathrm{z})] - \\ & \mathbb{E}_{q_\theta}[\log p_\phi(\mathrm{x}|\mathrm{z})] - \mathbb{E}_{q_\zeta^p}[\log p_\phi(\mathrm{x}|\mathrm{z})] \\ & - \mathbb{E}_{q_\zeta^p}[\log \frac{p(\mathrm{c}|\mathrm{z})}{p_\theta(\mathrm{x}|\mathrm{z})}] \end{aligned}$$

Lemma 2. $p(z|x,c) \approx E_{p(c)}[p(z|x,c)]$ *implies that* $p(c|z) \approx \frac{p(c)}{p(x)}p(x|z)$

Proof.

$$\begin{aligned} p(z|x,c) &\approx E_{p(c)}[p(z|x,c)] \\ p(c|z) &= p(z|c)p(c)/p(z) \\ p(x|z) &= p(z|x)p(x)/p(z) \\ \frac{p(c|z)}{p(x|z)} &\approx \frac{p(c)p(z|c,x)}{p(x)\mathbb{E}_{p(c)}[p(z|x,c)]} \\ \frac{p(c|z)}{p(x|z)} &\approx \frac{p(c)}{p(x)} \end{aligned}$$

□

How accurate the approximation is depends highly on on the relationship between $p(\mathrm{z}|\mathrm{x})$ and $p(\mathrm{z}|\mathrm{c})$.

Lemma 3. *In the case where $p(\mathrm{z}|\mathrm{x}) = p(\mathrm{z}|\mathrm{c})$, the approximation from 2 holds with equality.*

Proof. This can be easily seen by replacing the $p(\mathrm{z}|\mathrm{x})$ and $p(\mathrm{z}|\mathrm{c})$ in the proof of Lemma 2, we arrive to the same result without making the approximation step. □

This concludes the proof of proposition **??**.

Intuitively, $p(\mathrm{z}|\mathrm{x})$ and $p(\mathrm{z}|\mathrm{c})$ in the ideal case would be equal, i.e. knowing either x or c would be enough to obtain all the information about z, which allows for good information compression.

In our approach, we optimize $\mathbb{E}_{q_\zeta^p}[p_\phi(\mathrm{x}|\mathrm{z})]$ and $\mathbb{E}_{q_\theta}[p_\phi(\mathrm{x}|\mathrm{z})]$ by interchangebly sampling from p_ϕ and q_ζ^p based on a fair coin flip, ie Bernoulli distribution and assume an identity covariance Gaussian prior $p(\mathrm{z})$.

USER	hi . i was wondering if there are any eastern [value_food] type restaurant -s in the [value_area] part of the city?
SYSTEM	[restaurant_name] is a [value_food] restaurant in the [value_area] of town . it is [value_pricerange] -ly priced and located at [restaurant_address]. would you like to book a table?
USER	i am looking for an [value_pricerange] eastern [value_food] place in the [value_area]
SYSTEM	[restaurant_name] is a [value_pricerange] [value_food] restaurant in the [value_area] of town . would you like me to book a table for you?
USER	how about a [value_food] restaurant?
SYSTEM	i have [value_count] [value_pricerange] [value_food] restaurant -s in the [value_area] [restaurant_name], [restaurant_name], [restaurant_name], [restaurant_name], [restaurant_name], [restaurant_name], [restaurant_name], [restaurant_name], [restaurant_name], [restaurant_name], [restaurant_name], [restaurant_name], [restaurant_name], [restaurant_name], [restaurant_name], [restaurant_name], [restaurant_name], [restaurant_name]
USER	i would like a reservation for [value_count] to the [restaurant_name] restaurant.
SYSTEM	i have booked you a table for [value_count] at [restaurant_name] on [value_day] at [value_time]. your reference number is [restaurant_reference].
USER	i would like to make a reservation for [value_day] at [value_time] . and there has been a change in plans , i will be dining alone.
SYSTEM	your booking was successful . the table will be reserved for [value_count] minutes . reference number is [restaurant_reference].
USER	can i get the reference number for that reservation ?
SYSTEM	your booking was successful . the table will be reserved for [value_count] minutes . reference number is [restaurant_reference].
USER	excellent . that is everything!
SYSTEM	you are welcome . have a great day!

Table 4: Example of resulting delexicalized dialogue outputted by learned dialogue policy learned with optimal replay.

B LAVA Gauss Results

To obtain scores with the newest evaluation method for MultiWOZ, we ran the code available at https://gitlab.cs.uni-duesseldorf.de/general/dsml/lava-public. We didn't make any modification to the hyperparameters and ran the reinforcement learning stage for 10 epochs. Notably, the LAVA method with continuous latent exibits high variance in the success scores, with a standard deviation of 3.32 and mean success rate 73.33 across 3 runs, it is able to keep the BLEU score high since it's under-optimizing the success metric and is trained for few epochs which prevents divergence. In fact, TCUP can obtain the same BLEU score by sacrificing a bit on the success rate side, see Fig. 3b.

C Regularization Sensitivity

In Fig. 6 we do a regularization sensitivity analysis for different scales of KL regularization and fraction of optimal replay in the reinforcement learning training stage. We do the same for the categorical latent variable case in Fig. 7.

D Model and Training Setup

We make use of the Long-Short Term Memory (Hochreiter and Schmidhuber 1997) with a dot-product attention mechanism for the decoder architecture and Gated Recurrent Unit (Cho et al. 2014) for the encoder architecture. In both cases, we use a hidden size of 300 and a latent embedding vector of size 200. Note that in the case of prior-posterior shuffling, there are shared parameters for p_ϕ and q^p_ζ through the encoder, the hidden state is mapped to the appropriate mean and variance by independent neural networks. Under the context c we assume access to the ground-truth dialogue state, this can be easily replaced by the predictions of a dialogue state-tracker (LI et al. 2020; Lee et al. 2021).

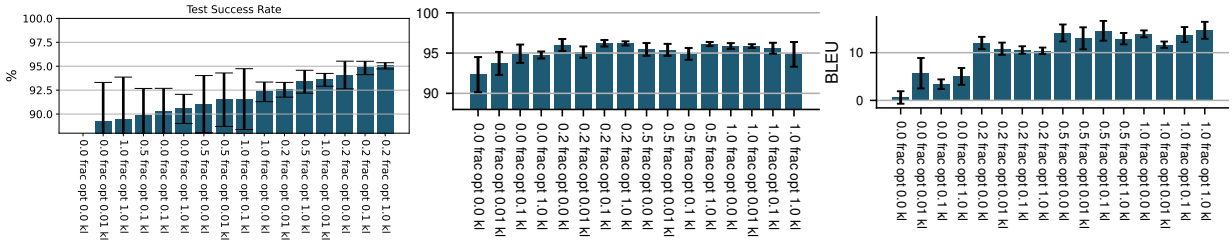
During training we choose the best models by keeping track of validation success rate and BLEU score and evaluate the best performing checkpoint on the test set. As discussed, the training of TCUP is separated into two stages.

D.1 Preprocessing

We follow the same approach as proposed by the MultiWOZ benchmark (Budzianowski et al. 2018) as well as prior work (e.g. (Lubis et al. 2020),) and delexicalize dialogues before running experiments. During the delexicalization of utterances slot values are replaced by corresponding type tokens. For instance, *I want to find a cheap restaurant...* is processed into *I want to find a [value_pricerange] restaurant....* More example utterances are listed in Tab. 4.

D.2 Success Metrics

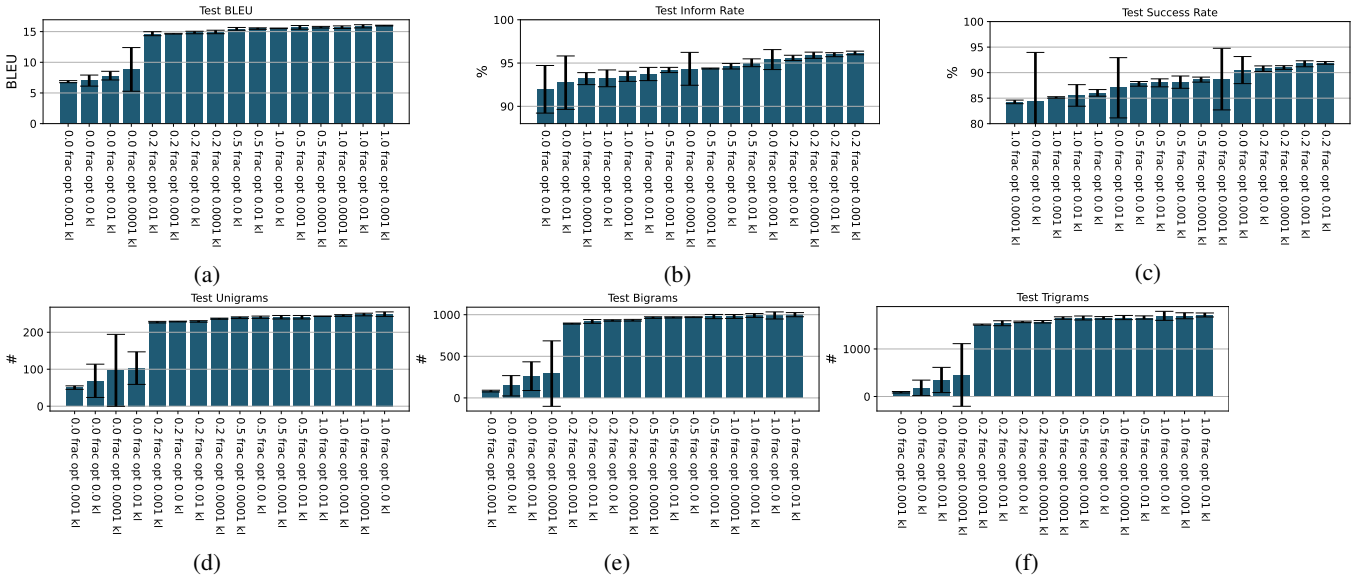
We used the default metric set for the context-to-response task as introduced by MultiWOZ (Budzianowski et al. 2018) to measure dialogue success. Each dialogue is associated with informable and requestable slots. Informable slots are attributes provided by the user to constrain the search (e.g.,



(a) (b) (c)

Figure 6: Regularization sensitivity 400 epochs.

area or price range). Requestable slots are additional information the users can request about a given entity (e.g., phone number). Consequently, inform rate computes whether informable slots are matched by the system, and success rate computes whether requested slots are provided by the system.



(a) (b) (c)

(d) (e) (f)

Figure 7: Sensitivity analysis for the categorical latent case with 10 30-dimensional categoricals in the latent.