

Learning Natural Language Generation with Truncated Reinforcement Learning

Alice Martin

Samovar, Télécom SudParis, Institut Polytechnique de Paris, Palaiseau
CMAP, École Polytechnique, Institut Polytechnique de Paris, Palaiseau

Guillaume Quispe and Charles Ollion

CMAP, École Polytechnique, Institut Polytechnique de Paris, Palaiseau

Sylvain Le Corff

Samovar, Télécom SudParis, Institut Polytechnique de Paris, Palaiseau

Florian Strub

Deep Mind

Olivier Pietquin

Google Brain

Abstract

This paper introduces TRUncated Reinforcement Learning for Language (TruFL), an original approach to train conditional language models without a supervised learning phase, by only using reinforcement learning (RL). As RL methods unsuccessfully scale to large action spaces, we dynamically truncate the vocabulary space using a generic language model. TruFL thus enables to train a language agent by solely interacting with its environment without any task-specific prior knowledge; it is only guided with a task-agnostic language model. Interestingly, this approach avoids the dependency to labelled datasets and inherently reduces pretrained policy flaws such as language or exposure biases. We evaluate TruFL on two visual question generation tasks, for which we report positive results over performance and language metrics, which we then corroborate with a human evaluation. To our knowledge, it is the first approach that successfully learns a language generation policy without pre-training, using only reinforcement learning.¹

1 Introduction

Since the development of generic language models trained on massive unlabelled text corpora (Radford et al., 2019; Brown et al., 2020), state-of-the art language processing systems rely on *sequential transfer learning* (Ruder, 2019). The pretrained Language Model (LM) is fine-tuned on the downstream task using a standard supervised learning (SL)

¹Code is available at <https://github.com/AMDonati/RL-NLP>

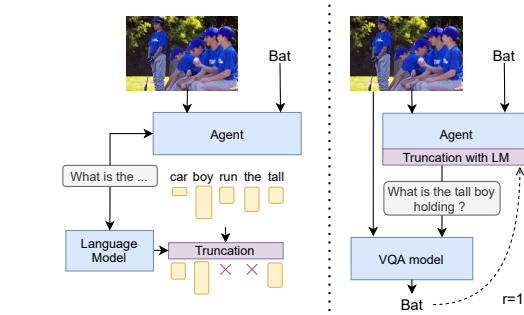


Figure 1: (left) In a conditional language generation task as VQG, TruFL truncates the vocabulary space by using a language model. Here, ‘run,’ and ‘the’ are syntactically incorrect and thus truncated. Yet, ‘car’ is not trimmed as the LM is not visually grounded. (right) In a VQG training loop, the agent generates a question given an image-answer pair, which is then fed to a VQA model predicting an expected answer. If both answers match, the agent is rewarded.

objective (Wu et al., 2019; Peters et al., 2019). Yet, such an approach suffers from several issues (Chen et al., 2020): (i) catastrophic forgetting when a model forgets previously learned knowledge and overfits to target domains, (ii) computational inefficiency from fine-tuning billion-parameters networks, and (iii) the need of supervised datasets. Moreover, task-specific language models learned with SL suffer from well-studied text degeneration issues (Holtzman et al., 2019), such as the exposure bias (Bengio et al., 2015), language biases (Saleh et al., 2020; Jaques et al., 2020), or a lack of diversity (Li et al., 2015).

On the other hand, text generation can be naturally framed as a sequential decision making problem, with the sequence of words seen as successive actions over a vocabulary. Thus, some researchers have recently focused on learning language models using instead

Reinforcement Learning (RL) (Strub et al., 2017; Das et al., 2017; Narasimhan et al., 2015). RL methods allow acquiring language through interactions within rich and diverse environments (Luketina et al., 2019), help understanding language acquisition and language pragmatics (Lazaridou et al., 2016; Bisk et al., 2020). "Reward is enough" (Silver et al., 2021) highlights the necessity of using RL for AI systems to acquire language in its full richness. Indeed, (i) language may be intertwined with other modalities of action and observation, (ii) the utility of language varies according to situations and behaviours, (iii) it is consequential and purposeful, and (iv) some linguistic problems are better solved dynamically, through experience (such as using a diplomatic tone in a speech.) In addition, RL allows optimizing a non-differentiable learning signal, hence handles more diverse objective functions, and also avoids some of the text degeneration issues previously mentioned. So far, RL-based text-generation tasks have relied on a pre-training phase to ease learning: the policy language model is trained with SL on the task dataset, before being fine-tuned with policy gradient methods (Sutton et al., 1999) on the task at hand. Those approaches often require human-labelled datasets. Besides, combining pre-training and fine-tuning phases either barely change the policy distribution, or induces *language drift* (Lazaridou et al., 2020; Lu et al., 2020b), i.e the generated language drifts semantically or syntactically from natural language.

In this paper, we aim at learning a conditional language model using *RL without a pre-training phase*, so that (i) we get free from datasets with human annotations, and (ii) we avoid the text generation flaws induced by the common methods. While appealing, such an approach requires overcoming the hurdle of the combinatorial language action space, a vocabulary usually containing more than 10,000 words. Yet, while large and discrete, a language action space contains a specific structure, made of all the syntactical and semantics rules of a given language. TruFLL leverages such structure to drive the exploration of the RL-based language agent during training. At each time step of the text generation process, TruFLL truncates its effective action space to a small subset of words provided by a pretrained task-agnostic language model. Such an approach injects a generic prior linguistic knowledge into the RL algorithm, is usable on tasks lacking in-domain labeled data, and can be easily transferred to new RL-based text generation tasks. Thus, TruFLL can

be applied to any language generation task given a generic LM and a reward. We here evaluate it on two Visual Question Generation (VQG) tasks, the synthetic CLEVR dataset (Johnson et al., 2017), and the natural language VQAv2 dataset (Goyal et al., 2017). Unlike alternative RL without pre-training approaches, TruFLL manages to ask meaningful and valid questions on large vocabularies, exhibiting success rate and language metrics close to pretrain models with labeled data, while producing more original language.

2 Background

Language Generation as an RL Problem. We cast the word-based text generation task as a Markov Decision Process to apply RL methods (Sutton et al., 1998). In this setting, a language model agent generates a sequence of words $w_{<t} = (w_0, w_1, \dots, w_{t-1})$ drawn from a vocabulary \mathcal{V} , given an initial context c associated with a reward r_t . Translation, text summarization or image captioning are examples of such tasks respectively using a source sentence, a text article, or an image as a context (c). During this process, the agent may be rewarded with language scores (Ranzato et al., 2016), human preferences (Stiennon et al., 2020) or task completion scores (Strub et al., 2017).

Formally, a language generation agent is defined by a policy π_θ (a distribution over \mathcal{V}) parametrized by θ , first initialized with the context c . At each time step t , the agent samples a new word w_t from its policy $\pi_\theta(w_t|w_{<t}, c)$. It moves to a new state $(w_{<t+1}, c)$ and receives a reward $r_t = r(w_{<t}, c, w_t)$, where r is a reward function relative to the language task. The RL language agent aims to learn a policy that maximizes $\mathbb{E}_{\pi_\theta}[\sum_{t=0}^T r_t]$,² while generating the sequence of words $w_{<T}$, where \mathbb{E}_{π_θ} is the expectation under π_θ , and T the maximal length of the words sequence.

Policy Gradient This optimization process may be performed through Policy Gradient (PG) algorithms (Sutton et al., 1999). In the language literature, REINFORCE (Williams, 1992) has been used as a simple Monte Carlo approximation of this gradient (Strub et al., 2017; Li et al., 2016). Yet, in this paper, we use a Proximal Policy Optimization approach (PPO) (Schulman et al., 2017) to have a lower variance and better convergence rate; PPO clips the gradient estimate to have smooth policy updates. For all $0 \leq t \leq T$, let $s_t = (w_{<t}, c)$ and $a_t = w_t$ be the state and action at time t . Policy gradient methods

²We cast the language modelling as an episodic problem with $\gamma = 1$ and omit the discount factor in the paper for clarity.

minimize the objective:

$$L_{\text{pg}}(\theta) = \mathbb{E}_{\pi_\theta} \left[\sum_{t=0}^T \log \pi_\theta(a_t | s_t) \hat{A}_t \right],$$

where \hat{A}_t is an estimator of the advantage function, here defined as $\hat{A}_t = \sum_{u=t}^T r_u - V_\phi(s_t)$ with $V_\phi(s)$ an estimator of the value function $V_{\pi_\theta}(s) = \mathbb{E}_{\pi_\theta}[\sum_{u=t}^T r(s_u, a_u) | s_t = s]$. PPO then keeps track of the previous policy $\pi_{\theta_{old}}$ before the PG update to compute the training objective:

$$L_{\text{ppo}}(\theta) = \mathbb{E}_{\pi_{\theta_{old}}} \left[\sum_{t=0}^T \rho_t^\theta \hat{A}_t \wedge \text{clip}(1-\epsilon, \rho_t^\theta, 1+\epsilon) \hat{A}_t \right],$$

where for all real numbers a, b , $a \wedge b = \min(a, b)$, $\rho_t^\theta = \pi_\theta(a_t | s_t) / \pi_{\theta_{old}}(a_t | s_t)$, ϵ is a hyper-parameter controlling the magnitude of the policy updates, and $\text{clip}(a, x, b)$ is the function that clips x in interval $[a, b]$. The expectation is estimated in practice using a Monte Carlo approach, with an empirical average over a finite batch of episodes, i.e. a succession of transitions $(s_t, a_t \sim \pi_{\theta_{old}}(\cdot | s_t), r_t)$ from an initial state s_0 to a terminal state s_T . Finally, the training loss is completed first with a value-based loss to learn the baseline V_ϕ that reduces the gradient variance; it computes for each timestep t of an episode the mean squared error $|\sum_{u=t}^T r_u - V_\phi(s_t)|^2$.³ Secondly, the loss is completed with an entropy term to soften the policy distribution, which computes for each timestep t of an episode $H(\pi_\theta(a_t | s_t))$, where H is the entropy function.

3 TruLL

We here aim at making RL methods feasible in the language setting by dynamically reducing the action space, i.e., by restricting the language agent to select a word within a subset of the vocabulary at each time step. We detail below the action space's truncation model and the associated RL algorithm to learn the language agent.

3.1 Dynamic Vocabulary Truncation

TruLL combines two distinct language models, which share the same vocabulary \mathcal{V} : a RL language agent π_θ and a pretrained language model f_{LM} . At each timestep t , TruLL restricts the vocabulary space of the RL language agent with:

$$\mathcal{V}_t^- = \{w | w \in \mathcal{V}, g_{trunc}(w | w_{<t}) = 1\},$$

³Note that other TD-based losses are applicable (Sutton et al., 1998; Schulman et al., 2016; Espeholt et al., 2018).

where g_{trunc} is a truncation function based on f_{LM} which either associates 0 or 1 with each word in the vocabulary given the past words $w_{<t}$. From a language modelling perspective, the vocabulary space of the language agent is reduced from \mathcal{V} to \mathcal{V}^- where $|\mathcal{V}^-| \ll |\mathcal{V}|$, with $|\cdot|$ the cardinal of a finite set. From a RL perspective, the RL agent follows a truncated policy π_θ^- which only samples actions over the subset \mathcal{V}^- . In practice, such a policy is computed using a masked softmax function over the truncated vocabulary \mathcal{V}_t^- : $\pi_\theta^-(\cdot | w_{<t}, c) = \text{softmax}(m * \text{logits}_{\pi_\theta}(w_{<t}, c))$ where $m=1$ when $g_{trunc}(w | w_{<t}) = 1$ otherwise $m=-\infty$.

3.2 Truncation Functions

We here list the different truncation functions g_{trunc} explored through the paper.

Top-k words: This function selects the k words with the highest probability given by $f_{LM}(\cdot | w_{<t})$:

$$g_{\text{top}(k)}(w_t | w_{<t}; k) = \mathbb{1}_{w_t \in \text{top}(k)(f_{LM}(\cdot | w_{<t}))}.$$

Probability threshold (α): This function only keeps words having a probability $f_{LM}(\cdot | w_{<t})$ greater than α :

$$g_{\text{Pth}(\alpha)}(w_t | w_{<t}; \alpha) = \mathbb{1}_{f_{LM}(w_t | w_{<t}) > \alpha}.$$

Top-p: This function is based on nucleus sampling (Holtzman et al., 2019), and it keeps the most likely words contained in a probability mass p of $f_{LM}(\cdot | w_{<t})$. Formally, we define \mathcal{V}_t^p as:

$$\mathcal{V}_t^p = \underset{|V_t|, V_t \subset \mathcal{V}}{\text{argmin}} \{w | w \in V_t, \sum_{w \in V_t} f_{LM}(w | w_{<t}) > p\},$$

and readily, $g_{\text{top}(p)}(w_t | w_{<t}; p) = \mathbb{1}_{w_t \in \mathcal{V}_t^p}$.

Sample (k): This function randomly samples k words from the language model with replacement to directly build the truncated vocabulary:

$$g_{\text{sample}(k)}(w_t | w_{<t}; k) = \mathbb{1}_{w_t \in \{w_i \sim f_{LM}(\cdot | w_{<t}) : i \in [1, \dots, k]\}}.$$

Only top(k) provides a fixed number of words at each time step. pth(α), top(p), and sample(k) have a dynamic truncation, whose size at t depends on the language model entropy.

3.3 Task-Specific vs. Generic LM

We benchmark two types of language models for truncation. On the one hand, we use an *external language model* pretrained on a large task-agnostic language corpora. Such a model provides a generic linguistic prior to the RL agent exploration process, solely encoding syntactic and semantic information. On the other hand,

we use a *task-related language model* pretrained on the supervised dataset associated with the task. Such a model provides a task-specific linguistic prior to the RL language agent, and captures language pragmatics. We emphasize that this paper aims at leveraging task-agnostic language models as they discard the need for task-specific data. For the sake of completeness, we also study the truncation with the task-related LM as an additional benchmark to assess our approach.

4 Experimental Setting

We here list the experimental setting and detail the network and hyperparameters in Appendix A.4.

4.1 Visual Question Generation

We showcase TruFLL on the task of Visual Question Generation (VQG) (Mostafazadeh et al., 2016), which is a form of Visual Jeopardy!™ (Ferrucci, 2012). There, the language agent observes an image-answer pair and has to generate a question that results in a similar answer, as illustrated in Figure 1. Such a task presents multiple advantages. First, by combining vision, scene understanding and language generation, it requires high-level reasoning and exhibits a large spectrum of language difficulties. Secondly, the success criterion is naturally non-differentiable, hence a natural fit for RL methods. Such a criterion, unlike metrics based on ground-truth sentences, allows generating diverse grounded questions given an image-answer pair.

Formally, the initial context c is composed of the image-answer pair $(\mathcal{I}, \mathcal{A})$. The RL agent then generates a sequence of words $w_{<t}$ of maximum length T . We then provide the generated question to a pretrained VQA model. This model takes as inputs the image \mathcal{I} , the generated question $w_{<t}$ and outputs a predicted answer $\hat{\mathcal{A}}$. Finally, the agent receives a reward $r(w_t, w_{<t}, c)$ based on \mathcal{A} and $\hat{\mathcal{A}}$.

4.2 Datasets

We evaluate TruFLL on the CLEVR and VQAv2 datasets to simulate large-scale VQG datasets. The two datasets have been originally created for the task of Visual Question Answering (VQA), i.e. for multi-modal classification algorithms predicting an answer given an image-question pair.

CLEVR The CLEVR VQA dataset (Johnson et al., 2017) is made of template questions on synthetic images, which contain simple objects with four distinct properties (shape, material, color, size). The vocabulary contains 86 words and 28 potential answers, making it a valuable proof of concept for assessing

TruFLL. Both language models are single-layer LSTMs (Hochreiter and Schmidhuber, 1997) with 512 units, and 512 word embedding dimension. The task-specific LM is trained over the full train dataset of CLEVR questions. The external language model is trained on the mixture of CLOSURE (Bahdanau et al., 2019) and CLEVR-Dialog (Kottur et al., 2019) datasets. Although those two datasets share the CLEVR vocabulary, their language distribution differs from vanilla CLEVR. Finally, we use a pretrained GT-Vector-NMN (Bahdanau et al., 2019) to compute the reward $r(w_t, w_{<t}, c) = \mathbb{1}_{\mathcal{A}=\hat{\mathcal{A}}, t=T-1}$, where $\mathbb{1}$ is the indicator function.

VQAv2 The VQAv2 dataset (Goyal et al., 2017) is made of natural language and open-formed questions on images from the MS-Coco Dataset (Lin et al., 2014). It has a vocabulary of 14,810 words and 3,149 answers. The task-specific language model is a one-layer LSTM with 512 units and a 512 word embedding dimension, pretrained over the full training dataset of VQAv2 questions. The External Language Model is Open-AI’s GPT-2 (Radford et al., 2019). The original language model outputs a probability distribution over 50,257 tokens, but we use a masked softmax function to restrict the probability distribution to the 14,810 tokens of the VQAv2 dataset. Unlike most NLP tasks relying on pretrained generic language models, we do not fine-tune it on the task dataset. Instead, we leverage the few-shot generalization capabilities of GPT-2, by feeding the language model with the prompt "Here are a few examples:" followed by 100 random questions $q_{<100}$ from the dataset. The truncation is then based on the probability distribution $f_{LM}^{gpt2}(\cdot | q_{<100}, w_{<t})$. Finally, we used a pretrained VilBERT to compute the reward (Lu et al., 2020a). Given the large number of answers, we use as reward a decreasing function of the rank of the reference answer $\text{rk}(\mathcal{A})$: $r(w_t, w_{<t}, c) = \mathbb{1}_{\text{rk}(\mathcal{A}) \leq 10, t=T-1} e^{-\text{rk}(\mathcal{A})/2}$, as further explained in Appendix A.5.

In these two settings, we acknowledge that the task dataset is still used to train the VQA models. Please note that the VQA modules are only used to model the environment, i.e. to provide a positive/negative feedback to the agent. In other settings, TruFLL would still work if we replace the VQA model by any language interface: text-game (e.g. Zork), expert-systems, or humans. Here, we only use the VQG framework as a proof of concept that natural language can be learned *through pure interaction* given any task reward. Other language generation applications are discussed in Section 5.3.

4.3 Baselines

In this paper, we aim to show that a RL language agent can be trained from scratch, i.e. without the usual pre-training phase by solely interacting with another language system, the VQA model, when supported by truncation methods. The truncation with the task-related LM is referred to as TruFLL (Task-LM), while the one with the External LM is referred as TruFLL (Ext-LM). We first emphasize the difficulty of training an RL language agent without a supervised pre-training phase through two baselines. We trained a simple on-policy PPO algorithm without any action space pruning, and refer to it as *scratch*. Then, we added a Kullback-Leibler (KL) regularization term to the loss, $\lambda_{\text{KL}} \text{KL}(\pi_\theta || f_{LM})$, with $\lambda_{\text{KL}} > 0$, to incorporate language prior to the agent as in (Jaques et al., 2017, 2019). We refer to it as *scratch + KL-task* when distilling the task-specific language model, and *scratch + KL-ext* with the external language model. Finally, we include two baselines with a pre-training phase. We trained a language agent on the task-dataset with a log-likelihood objective, and refer to it as *pretrain*. Then, we fine-tune the pretrained language agent with PPO without truncation, and refer to it as *pretrain + RL fine-tune*. These two baselines should be viewed as gold standards as they rely on task-related data; additionally, *pretrain + RL fine-tune* is today the state-of-the-art method for learning RL-based LM.

4.4 Metrics and Evaluation Methods

Evaluating text generation is an open-research problem in language literature. We decompose automatic language evaluation into three categories to assess different facets of language, and perform as well a human evaluation study.

Performance metrics. We measure the task-completion score or recall @ 1 which states whether the target answer \mathcal{A} is the top answer of the VQA models, and the recall @ 5 (R@5), which assesses whether \mathcal{A} is in the 5 top answers. These scores measure the task-solving abilities of the agent, but they are also conditioned by the VQA model abilities.

Language Metrics. First, we used n-grams metrics, BLEU (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005) and CIDEr (Vedantam et al., 2015), to measure the similarity between the generated question and the reference questions in the evaluation set. While those scores can capture syntactic and semantic properties of language, they also fall short when dealing with open-form language, e.g. an

identical answer may arise from two non-overlapping but syntactically correct questions. Thus, we also compute two metrics assessing the quality of the language independently of reference questions, the perplexity of the question given an external LM (ppl-e), and its perplexity given the task-related LM (ppl-t).

Diversity Metrics. We here estimate a self-BLEU (sBLEU) score (Zhang et al., 2017) over 10 questions generated on the same image-answer pair. Although such score detects potential mode collapse, i.e., when the language utters identical sequences of words, it also values babbling, i.e., outputting random words. We thus also measure the probability mass of the ten most frequent words (Choshen et al., 2020), and refer to it as peakiness (peak).

Human Evaluation. On the VQAv2 task, we also performed human evaluation by surveying 53 participants on the first 50 questions produced by some of the models at test time. The study (further detailed in Appendix C) is based on pairwise comparison of question samples produced by the concurrent algorithms according to four criteria. First, we evaluated the language quality of the question samples, by asking the participants to select the most syntactically and semantically correct question among the two samples of the questions pair. Secondly, we evaluated language grounding, i.e adequacy of the sample to the image-answer pair, by asking the participants to select the question most suitable given the two elements. Thirdly, we evaluated the language originality and diversity, by asking participants to select the question the most different from the dataset reference question. Finally, we evaluated the number of syntax errors by asking participants to tick the question if it is grammatically incorrect. Examples of questions asked during the study are included in the Appendix C.

4.5 Sampling methods for text generation

When generating text from a trained language model, the quality and diversity of samples depend on the decoding algorithm (Zhang et al., 2020). We consider three text generation methods. *greedy* uses the argmax of the policy, while *sampling* uses the multinomial distribution. Finally, we sampled ten text sequences from the policy, and selected the one with the lowest perplexity according to the external language model, and refer to it as *lm-ranking*. This process has been used recently in Text-to-Image Generation tasks (Ramesh et al., 2021).

Method	Score	R@5	BLEU	Meteor	CIDEr	ppl-t (\downarrow)	ppl-e (\downarrow)	sBLEU (\downarrow)	peak. (\downarrow)
Pretrain	0.30	0.71	0.19	0.38	0.83	3.1	31	0.44	0.96
Pretrain + RL fine-tune	0.44	0.86	0.17	0.34	0.70	4.0	35	0.46	0.95
Scratch	0.17	0.47	0.05	0.08	0.10	10^9	10^6	0.14	0.26
Scratch + KL-task	0.14	0.38	0.15	0.30	0.53	92	10^2	0.34	0.94
Scratch + KL-ext	0.17	0.44	0.14	0.27	0.43	10^4	28	0.37	0.95
TruFL (Task-LM)	0.56	0.90	0.17	0.32	0.66	3.4	23	0.95	1.00
TruFL (Ext-LM)	0.48	0.93	0.08	0.18	0.34(± 0.10)	10^3	3.0	0.95	1.00

Table 1: **CLEVR metrics** on 5k test episodes with 50k train episodes on 20k Images. Scores are averaged over the three decoding procedures mentioned in Section 4.5 and over 5 seeds; standard deviations are displayed when greater than 0.01 for accuracy metrics. We here report the models with the highest task-success:, i.e. the *scratch+KL* baselines with $\lambda_{KL} = 0.1$, and the truncation model with a probability threshold, $p_{th}(\alpha=0.05)$. Best values are underlined, best values without task-data (from scratch) are in bold.

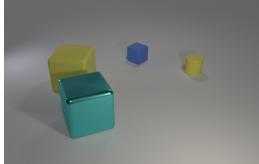
	Human	There is a blue thing that is the same shape as the big cyan metallic object ; what is its size? A:Small
	pretrain	There is a red metallic object that is the same size as the yellow rubber block ; what is its size?
	pretrain + RL	What size is the thing that is the same color as the matte cube ? <input checked="" type="checkbox"/>
	scratch	size sphere small blue or a yellow green large else in cylinders cubes color and how matte objects cube
	scratch+KL-task	How big is the shiny cylinder ?
	scratch+KL-ext	How many other objects in the are of same color as that shiny object ?
	TruFL (Task-LM)	How big is the thing that is to the right of the big matte thing ? <input checked="" type="checkbox"/>
	TruFL (Ext-LM)	What is the size of the thing that is right of the big cyan thing and is the same shape? <input checked="" type="checkbox"/>
	Human	What color is the cat A:Black
	pretrain	What color is the cat's collar? <input checked="" type="checkbox"/>
	pretrain + RL	What color is the cat? <input checked="" type="checkbox"/>
	scratch	AmazingAmazingAmazingAmazingAmazingAmazing
	scratch+KL-task	What color is their hat of the fingers of this?
	scratch+KL-ext	The first time is a bit of the way
	TruFL (Task-LM)	What color is her outfit? <input checked="" type="checkbox"/>
	TruFL (Ext-LM)	What color can these cats look like in real life? <input checked="" type="checkbox"/>

Figure 2: Samples on CLEVR and VQA: the checkbox indicates that the question generates the correct answer.

5 Results

5.1 CLEVR results

Quantitative performance: In Table 1, vanilla RL from scratch fails to have a decent performance even with synthetic language. Besides, adding a KL regularisation term does kick-start the learning process. Yet, as soon as we apply the dynamic truncation, TruFL matches the pretrained baselines performance when using the external LM, and even outperforms them with the task-specific LM. In this synthetic VQG setting, TruFL seems to be a viable and promising procedure to learn a RL language agent without a supervised training phase. Pretrained baselines have high language scores when assessed with dataset-based metrics, e.g BLEU or task-perplexity. Yet, they also remain close to the original dataset distribution with a medium external perplexity. Noticeably, TruFL with the task-specific LM follows the same pattern. On the other hand, TruFL with the external LM reports poor dataset-based language scores, while maintaining a low external perplexity. Therefore, TruFL seems to correctly capture the language distribution of the initial LM. As the performance score is high when using an external LM, it suggests that our approach can learn a policy on a language task with-

out the need of a task-related dataset. Less positively, TruFL diversity metrics suggest potential mode collapse, with a high peakiness and self-BLEU score.

Qualitative performance: We display qualitative samples in Figure 2 and Appendix D. On the one hand, the pretrained baselines generate either a question inconsistent with the visual context, or which fails to answer the expected answer. They inaccurately capture the pragmatics of the task. On the other hand, TruFL generate adequate questions, resulting in the expected answer. Interestingly, they are often grounded with different objects of the image. It is remarkable that TruFL with a generic LM still manages to capture the necessary subtleties of VQG, without any prior task knowledge. Despite a peaky distribution, TruFL has moderate repetitions across images, and is mostly over-confident. As for the scratch+KL samples, they are either not grounded, or showcase degenerated language.

Truncation function in CLEVR: In Table 2, we evaluate the different truncation functions defined in Section 3. While all truncation methods report similar task performance, the dynamic truncation functions, i.e. $p_{th}(\alpha)$, $top(p)$ and $sample(k)$, outperform the $top(k)$ regarding language metrics. Interestingly, the $sample(k)$ one, which generates a stochastic truncated

Trunc.	Score	BLEU	CIDEr	ppl-e(\downarrow)	sBLEU(\downarrow)
TruFL (Task-LM)					
top(k)	0.50	0.12	0.32	100	0.93
$p_{th}(\alpha)$	0.54	0.17	0.65	24	0.95
top(p)	0.51	0.17	0.69	12	0.96
sample(k)	0.50	0.18	0.73	16	0.89
TruFL (Ext-LM)					
top(k)	0.52	0.06	0.15	151	0.94
$p_{th}(\alpha)$	0.48	0.08	0.34(± 0.10)	3.0	0.95
top(p)	0.45	0.10	0.40(± 0.17)	3.3	0.92
sample(k)	0.41	0.13	0.46(± 0.16)	2.7	0.92

Table 2: **CLEVR task:** Truncation functions with parameters: top($k=10$), $p_{th}(\alpha=0.05)$ top($p=0.85$), sample($k=20$). Best values are underlined, best values for each TruFL algorithms are in bold.

action space, while having a lower performance, yields to the most correct and diverse language, with higher language scores and a lower self-BLEU. A stochastic action space might be harder to explore efficiently for reaching good task-solving abilities, but might strengthen the agent language generation properties.

5.2 VQAv2 task

In CLEVR, we observe that TruFL seems a promising approach to learn a language policy without a supervised training phase, by solely interacting with another language system. We scale our approach to natural language with large vocabulary (15k tokens) through the VQAv2 dataset.

Quantitative performance: Table 3 reports the VQAv2 results, for which TruFL and the baselines present a similar trend than on CLEVR. First, the scratch baselines keep failing to learn a valuable policy, with performance scores and n-grams metrics close to zero. Although TruFL does not outperform the performance of the pretrained baselines anymore, it still leads to similar performances, and satisfactory language scores. The similarity between TruFL (Task-LM) and TruFL (Ext-LM) results suggests that the truncation approach is viable when using a generic LM whose original vocabulary distribution differs from the task. Interestingly, TruFL displays a self-BLEU score similar to the pretrained baselines. This suggests that the poor diversity behavior observed on CLEVR is likely attributable to the small vocabulary and synthetic language distribution.

Qualitative performance: In Figure 2 and Appendix D, we display question samples for all models. TruFL and the pretrained baselines successfully generate a question giving the expected answer ("Black"), while the RL from scratch baselines fail, and even showcase degenerated language. Pretrained baselines tend to output a question closer to the reference question whereas TruFL outputs original

questions which differs from the VQA distribution, yet consistent with the context.

Human Evaluation: Figure 3 details the Human Evaluation results. Among the RL from scratch baselines, we selected scratch+KL-task as the only model producing sometimes meaningful questions. Yet, it fails to generate correct and grounded language; it is thus not a viable approach despite its diverse output. In line with the automatic metrics, the supervised baselines produce the best language, while being accurately grounded. Yet, they exhibit significantly less diversity with the reference language; this suggests in particular that pretrain+RL fails to go beyond the initial task-data distribution. Finally, unlike TruFL (Task-LM) which suffers from syntactic errors, TruFL (Ext-LM) produces language that qualitatively competes with pretrain models (53%), with a similar ratio of syntactic uncorrect samples. Although its questions are less grounded, they are diverse, which suggests that they follow a different distribution from the initial VQA dataset. It confirms that TruFL (Ext-LM) could be an alternative approach as it has an excellent trade-off between language quality, diversity, and grounding.

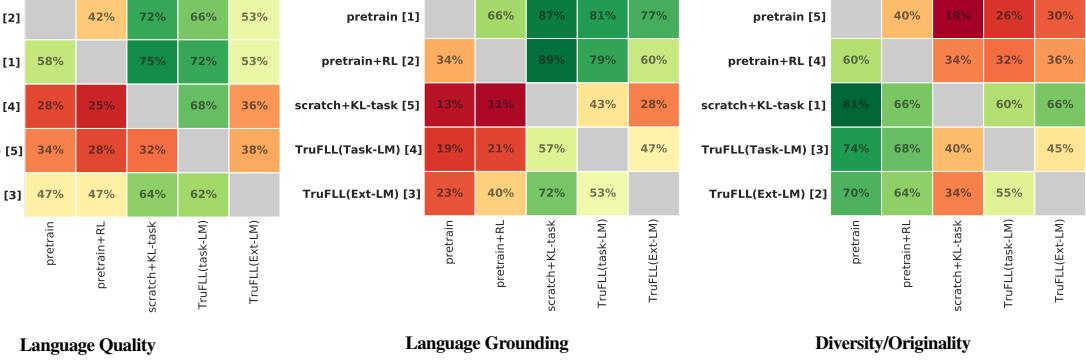
Decoding procedure: In Table 4, we evaluate the text sampling procedures described in Section 4.5. While greedy decoding produces the best outcome for pretrained models, lm-ranking provides an excellent trade-off between task performance and language quality with RL-based methods. As PG solely optimizes the task success ratio, this may reduce overall language quality, the re-ranking thus retrieves the best syntactically sentences a posteriori.

5.3 Discussion

Removing the truncation at evaluation with off-policy RL. So far, TruFL directly learns the truncated policy over the truncated vocabulary \mathcal{V}_t^- in an on-policy scheme. Hence, the algorithm requires the truncation, and a fortiori the language model, at test time. In this section, we investigate if we can directly learn a policy over the full vocabulary, and thus removing the truncation at test time. In such a setting, we adopt an off-policy training scheme, where the trajectories used to learn the behavior π_θ at training time are sampled under a different policy, the truncated policy π_θ^- . Thus, we need to unbiased the PG by using an importance sampling term between the exploratory policy π_θ^- and the behavior policy π_θ (Degris et al.,

Method	Score	R@5	BLEU	Meteor	CIDEr	ppl-t (↓)	ppl-e (↓)	sBLEU (↓)	peak.(↓)
Pretrain	0.38	0.59	0.30	0.40	0.93	<u>12</u>	<u>24</u>	0.80	0.99
Pretrain + RL fine-tune	<u>0.41</u>	<u>0.63</u>	<u>0.31</u>	<u>0.41</u>	<u>0.98</u>	21	50	0.78	0.99
Scratch	0.01	0.04	0.00	0.00	0.00	10^7	10^6	0.75	1.00
Scratch + KL-task	0.11	0.29	0.24	0.27	0.24	10^2	10^2	0.27	0.74
Scratch + KL-ext	0.01	0.05	0.06	0.04	0.01	10^6	10^3	0.10	0.20
TruFLL (Task-LM)	0.35	0.56	0.21	0.15	0.11	24	10^2	0.78	0.99
TruFLL (Ext-LM)	0.34	0.52	0.18	0.15	0.04	10^2	24	0.83	0.99

Table 3: **VQAv2** metrics on 20k test episodes with 100k train episodes. Scores are averaged over the three decoding procedures. *scratch+KL* has $\lambda_{KL}=0.05$, the truncation for TruFLL with (Task-LM) and TruFLL (Ext-LM) are respectively $p_{th}(\alpha=0.005)$ and $P_{th}(\alpha=0.0075)$. Best values are underlined, best values without task-data are in bold.



Pairwise comparisons: % of questions chosen for the model in bold (rows) when compared to the concurrent model (columns).

Syntax errors	pretrain (2)	pretrain+RL (3)	scratch+KL-task (5)	TruFLL (Task-LM) (4)	TruFLL (Ext-LM) (1)
	16%	17%	27%	24%	15%

Figure 3: **VQAv2** results for Human Evaluation study detailed in Section 4.4. The three matrices on top are pairwise comparisons: each cell displays the proportion of questions chosen for the models in the row (bold) when compared to the concurrent model in the column. The table at the bottom displays the proportion of incorrect questions coming from each model among all incorrect samples. In all figures, bracket numbers indicates the model rank per criteria, from 1="best" to 5="worst".

Method	Text-gen	Score	BLEU	CIDEr	ppl-e
pretrain	greedy	0.40	<u>0.32</u>	1.01	51
	sampling	0.37	0.30	0.88	62
	lm-ranking	0.37	0.14	0.87	54
pretrain + RL	greedy	0.42	0.32	1.05	55
	sampling	0.40	0.30	0.92	71
	lm-ranking	0.40	0.31	0.99	26
TruFLL (Task-LM)	greedy	0.36	0.20	0.11	366
	sampling	0.35	0.20	0.11	337
	lm-ranking	0.34	0.21	0.11	95
TruFLL (Ext-LM)	greedy	0.36	0.18	0.04	25
	sampling	0.34	0.18	0.04	28
	lm-ranking	0.33	0.19	0.15	20

Table 4: **VQAv2**: Ablation on the sampling methods. Overall best values are underlined, TruFLL best values are in bold.

2012). Formally, the off-policy PPO loss is defined by:

$$L_{\text{ppo}}^{\text{off}}(\theta) = \mathbb{E}_{\pi_{\theta}^-} [\min(\bar{\rho}_t^{\theta} A_t, \text{clip}(1-\epsilon, \bar{\rho}_t^{\theta}, 1+\epsilon) A_t)],$$

where $\bar{\rho}_t^{\theta} = \frac{\pi_{\theta}(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)} \frac{\pi_{\theta_{old}}(a_t|s_t)}{\pi_{\theta_{old}}^-(a_t|s_t)}$ is the new ratio.⁴

Table 5 displays the on-policy and off-policy results on both VQG tasks for TruFLL (task-LM), and is further detailed in Appendix B.3. We also

⁴Note that we did not simplify the expression to highlight the importance sampling ratio.

monitor the probability mass of the policy attributed to the truncated action space (sumVA). The policy only samples words within the truncated action space when sumVA = 1, without needing the truncation. On CLEVR, the TruFLL_{off} has lower - yet close - performance on language and task scores than TruFLL. As its sumVA ratios are very close to 1, the agent has learned to generalize over the full vocabulary. However, the approach does not manage to sufficiently scale to VQAv2. It could be improved with regularisation techniques and the use of TruFLL within state-of-the-art off-policy RL algorithms. We leave such possibilities to future works.

Algo	Score	BLEU	CIDEr	ppl-e	sBLEU	sumVA
CLEVR						
TruFLL	0.56	0.17	0.06	10^3	0.78	N.A
TruFLL _{off}	0.50	0.14	0.43	10^4	0.88	0.96
VQAv2						
TruFLL	0.35	0.21	0.11	10^4	0.36	N.A
TruFLL _{off}	0.07	0.03	0.01	10^4	0.05	0.08

Table 5: On-policy vs. off-policy scores: when training with an off-policy loss, we remove the truncation at test time.

Additional experiments. We sweep over truncation hyper-parameters in Table 6 of Appendix B. In Table 8, we observe that rewarding an agent with a BLEU score is sub-optimal in both language and task scores on CLEVR. In VQA, we apply temperature scheduling on the LM to perform fine-grained truncations in Table 9 of B.2. Finally, we explore TruflL with a pre-training phase in Table 10.

Generalization of the approach. TruflL learns conditional language models able to solve specific Natural Language Generation tasks given a context c . For solving such tasks, it only requires the context, a reward function that scores the language generated by the RL agent with respect to the task, and eventually a few natural language demonstrations fed as input prompt to the generic language model used in the truncation algorithm. Hence, the method is transferable to a wide variety of NLG tasks, without requiring upfront large-scale labelled datasets. Additionally, the RL framework allows to optimize non-differentiable objectives, making TruflL a natural choice to learn end-to-end task-oriented dialogs, such as (De Vries et al., 2017; Das et al., 2017). Other interesting tasks for TruflL include the ones typically found in Vision and Language Representation Learning (Lu et al., 2020a), such as Image Captioning, Grounding Referring Expressions (generation of a referring expression over a specific bounding box of an image), Caption-based Image Retrieval (generation of a caption that discriminates an image between a set of images). Reward functions for such tasks can be based on similarity scores between the generated language and the associated image or image region, which can be computed using pretrained language representations such as BERT (Devlin et al., 2019) or multi-modal pretrained systems such as ViLBERT (Lu et al., 2019). The context can be any kind of data structure (natural language, database, video, etc): if it is a linguistic input, TruflL can be applied for instance to text summarization, paraphrase generation (with reward functions based on similarity scores between the context and the generated language) or text-based games (Ammanabrolu and Riedl, 2018).

6 Related work

RL and NLP Tasks. Following (Singh et al., 2002; Lemon and Pietquin, 2007), recent RL-based task-oriented dialogues (De Vries et al., 2017; Das et al., 2017; Lewis et al., 2017; Narasimhan et al., 2015) have been developed, where the policy language model is generally pretrained with SL followed RL

fine-tuning. Yang et al. (2018); Fan et al. (2018) focused on tackling VQG tasks with RL, respectively on CLEVR and on the VQG dataset. Yet, the former uses slot filling with template questions, while the latter computes a mixed objective with a MLE loss using ground-truth sentences. Bahdanau et al. (2016); Rennie et al. (2017) use RL to train language models as an alternative to SL to prevent typical text degeneration issues, but within training algorithms relying on ground-truth examples from labelled datasets.

RL methods for Language Action Spaces. Several RL algorithms have been developed to tackle large discrete action spaces. Hence, Dulac-Arnold et al. (2015); Tennenholz and Mannor (2019); Chandak et al. (2019) embed the actions into a continuous action space, and then use classic RL algorithms to learn a policy over this continuous space. Zahavy et al. (2018); Seurin et al. (2020) proposes Q-learning algorithms with an elimination signal to eliminate forbidden actions. Closer to our work, a few algorithms (Ammanabrolu and Riedl, 2018) use the structure of language to prune the action space of text-based games, but within value-based algorithms, which are less scalable to large vocabularies. Similarly to TruflL, CALM (Yao et al., 2020) combines a pretrained language model to prune the action space with a Deep-Q network, aka DRNN (He et al., 2016). Yet, its truncation language model remains fine-tuned on the RL dataset. Besides, CALM is only evaluated on a vocabulary of 697 tokens, and on 4-words action sequences.

Learning Language Models from scratch. (Ziegler et al., 2019; Garg et al., 2021) finetune pretrained GPT-2 models with RL for language generation tasks without task-related data, only using reward signals. Yet, they still face optimization and computational challenges (Parisotto et al., 2020).

7 Conclusion

We proposed TruflL, an original approach to learn a natural language generation (NLG) task using RL, without the usual pre-training phase requiring supervised datasets. To our knowledge, this is the first RL-based algorithm dedicated to learning a word-based text-generation task, which does not rely on a pre-training phase while scaling to large vocabularies. Although it comes with its limitations, the truncated RL algorithm provided by TruflL gets free from labelled data in task-oriented language models, presents interesting language generation properties, and provides a generic and transferable method to learn any NLG problem.

Acknowledgements

This action benefited from the support of the Chair New Gen RetAll led by l’X – École Polytechnique and the Fondation de l’Ecole Polytechnique, sponsored by CARREFOUR.

References

- Prithviraj Ammanabrolu and Mark O Riedl. 2018. Playing text-adventure games with graph-based deep reinforcement learning. In *Proc. of the North American Chapter of the Association for Computational Linguistics (NAACL)*.
- Dzmitry Bahdanau, Philemon Brakel, Kelvin Xu, Anirudh Goyal, Ryan Lowe, Joelle Pineau, Aaron Courville, and Yoshua Bengio. 2016. An actor-critic algorithm for sequence prediction. In *Proc. of the International Conference on Learning Representations (ICLR)*.
- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proc. of the International Conference on Learning Representations (ICLR)*.
- Dzmitry Bahdanau, Harm de Vries, Timothy J O’Donnell, Shikhar Murty, Philippe Beaudoin, Yoshua Bengio, and Aaron Courville. 2019. Closure: Assessing systematic generalization of clevr models. *Visually Grounded Interaction and Language (ViGIL)*.
- Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proc. of ACL Workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*.
- Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. In *Proc. of Neural Information Processing Systems (NIPS)*.
- Yonatan Bisk, Ari Holtzman, Jesse Thomason, Jacob Andreas, Yoshua Bengio, Joyce Chai, Mirella Lapata, Angeliki Lazaridou, Jonathan May, Aleksandr Nisnevich, Nicolas Pinto, and Joseph Turian. 2020. Experience grounds language.
- Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. In *Proc. of Neural Information Processing Systems (NeurIPS)*.
- Yash Chandak, Georgios Theodoros, James Kostas, Scott Jordan, and Philip S. Thomas. 2019. Learning action representations for reinforcement learning. In *Proc. of International Conference on Machine Learning (ICML)*.
- Sanyuan Chen, Yutai Hou, Yiming Cui, Wanxiang Che, Ting Liu, and Xiangzhan Yu. 2020. Recall and learn: Fine-tuning deep pretrained language models with less forgetting. In *Proc. of Empirical Methods in Natural Language Processing (EMNLP)*.
- Leshem Choshen, Lior Fox, Zohar Aizenbud, and Omri Abend. 2020. On the weaknesses of reinforcement learning for neural machine translation. In *Proc. of the International Conference on Learning Representations (ICLR)*.

- Abhishek Das, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, José MF Moura, Devi Parikh, and Dhruv Batra. 2017. Visual dialog. In *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Harm De Vries, Florian Strub, Sarath Chandar, Olivier Pietquin, Hugo Larochelle, and Aaron Courville. 2017. Guesswhat?! visual object discovery through multi-modal dialogue. In *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Thomas Degris, Martha White, and Richard S Sutton. 2012. Off-policy actor-critic. *arXiv preprint arXiv:1205.4839*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. **BERT: Pre-training of deep bidirectional transformers for language understanding.** 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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Gabriel Dulac-Arnold, Richard Evans, Hado van Hasselt, Peter Sunehag, Timothy Lillicrap, Jonathan Hunt, Timothy Mann, Theophane Weber, Thomas Degris, and Ben Coppin. 2015. Deep reinforcement learning in large discrete action spaces. *arXiv preprint arXiv:1512.07679*.
- Lasse Espeholt, Hubert Soyer, Remi Munos, Karen Simonyan, Vlad Mnih, Tom Ward, Yotam Doron, Vlad Firoiu, Tim Harley, Iain Dunning, et al. 2018. Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures. In *Proc. of International Conference on Machine Learning (ICML)*.
- Zhihao Fan, Zhongyu Wei, Siyuan Wang, Yang Liu, and Xuan-Jing Huang. 2018. A reinforcement learning framework for natural question generation using bi-discriminators. In *Proc. of the International Conference on Computational Linguistics (ICCL)*.
- David A Ferrucci. 2012. Introduction to “this is watson”. *IBM Journal of Research and Development*, 56(3.4):1–1.
- Sonal Garg, Sumanth Prabhu, Hemant Misra, and G Srinivasaraghavan. 2021. Unsupervised contextual paraphrase generation using lexical control and reinforcement learning. *arXiv preprint arXiv:2103.12777*.
- Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Ji He, Jianshu Chen, Xiaodong He, Jianfeng Gao, Lihong Li, Li Deng, and Mari Ostendorf. 2016. Deep reinforcement learning with a natural language action space. In *Proc. of Association for Computational Linguistics (ACL)*.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.
- Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. In *Proceedings of the International Conference on Learning Representations (ICLR)*.
- Eric Jang, Shixiang Gu, and Ben Poole. 2016. Categorical reparameterization with gumbel-softmax. In *Proc. of the International Conference on Learning Representations (ICLR)*.
- Natasha Jaques, Asma Ghandeharioun, Judy Hanwen Shen, Craig Ferguson, Agata Lapedriza, Noah Jones, Shixiang Gu, and Rosalind Picard. 2019. Way off-policy batch deep reinforcement learning of implicit human preferences in dialog. *arXiv preprint arXiv:1907.00456*.
- Natasha Jaques, Shixiang Gu, Dzmitry Bahdanau, José Miguel Hernández-Lobato, Richard E Turner, and Douglas Eck. 2017. Sequence tutor: Conservative fine-tuning of sequence generation models with kl-control. In *Proc. of International Conference on Machine Learning (ICML)*.
- Natasha Jaques, Judy Hanwen Shen, Asma Ghandeharioun, Craig Ferguson, Agata Lapedriza, Noah Jones, Shixiang Gu, and Rosalind Picard. 2020. Human-centric dialog training via offline reinforcement learning. In *Proc. of Empirical Methods in Natural Language Processing (EMNLP)*.
- Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. 2017. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*.
- Satwik Kottur, José M. F. Moura, Devi Parikh, Dhruv Batra, and Marcus Rohrbach. 2019. CLEVR-dialog: A diagnostic dataset for multi-round reasoning in visual dialog. In *Proc. of the North American Chapter of the Association for Computational Linguistics (NAACL)*.
- Angeliki Lazaridou, Alexander Peysakhovich, and Marco Baroni. 2016. Multi-agent cooperation and the emergence of (natural) language. *CoRR*, abs/1612.07182.
- Angeliki Lazaridou, Anna Potapenko, and Olivier Tieleman. 2020. Multi-agent communication meets natural language: Synergies between functional and structural language learning. In *Proc. of Association for Computational Linguistics (ACL)*.
- Oliver Lemon and Olivier Pietquin. 2007. Machine learning for spoken dialogue systems. In *Proc. of Conference of the International Speech Communication Association (ISCA)*.
- Mike Lewis, Denis Yarats, Yann N Dauphin, Devi Parikh, and Dhruv Batra. 2017. Deal or no deal? end-to-end learning for negotiation dialogues. In *Proc. of Empirical Methods in Natural Language Processing (EMNLP)*.

- Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2015. A diversity-promoting objective function for neural conversation models. In *Proc. of the North American Chapter of the Association for Computational Linguistics (NAACL)*.
- Jiwei Li, Will Monroe, Alan Ritter, Michel Galley, Jianfeng Gao, and Dan Jurafsky. 2016. Deep reinforcement learning for dialogue generation. In *Proc. of Empirical Methods in Natural Language Processing (EMNLP)*.
- Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *Proc. of European Conference on Computer Vision (ECCV)*.
- Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*.
- Jiasen Lu, Vedanuj Goswami, Marcus Rohrbach, Devi Parikh, and Stefan Lee. 2020a. 12-in-1: Multi-task vision and language representation learning. In *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Yuchen Lu, Soumey Singh, Florian Strub, Olivier Pietquin, and Aaron Courville. 2020b. Countering language drift with seeded iterated learning. In *Proc. of International Conference on Machine Learning (ICML)*.
- Jelena Luketina, Nantas Nardelli, Gregory Farquhar, Jakob Foerster, Jacob Andreas, Edward Grefenstette, Shimon Whiteson, and Tim Rocktäschel. 2019. A survey of reinforcement learning informed by natural language. In *Proc. of International Joint Conferences on Artificial Intelligence (IJCAI)*.
- Nasrin Mostafazadeh, Ishan Misra, Jacob Devlin, Margaret Mitchell, Xiaodong He, and Lucy Vanderwende. 2016. Generating natural questions about an image. In *Proc. of Association for Computational Linguistics (ACL)*.
- Karthik Narasimhan, Tejas Kulkarni, and Regina Barzilay. 2015. Language understanding for text-based games using deep reinforcement learning. In *Proc. of Empirical Methods in Natural Language Processing (EMNLP)*.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proc. of the Association for Computational Linguistics (ACL)*.
- Emilio Parisotto, Francis Song, Jack Rae, Razvan Pascanu, Caglar Gulcehre, Siddhant Jayakumar, Max Jaderberg, Raphael Lopez Kaufman, Aidan Clark, Seb Noury, et al. 2020. Stabilizing transformers for reinforcement learning. In *Proc. of International Conference on Machine Learning (ICML)*.
- Matthew E Peters, Sebastian Ruder, and Noah A Smith. 2019. To tune or not to tune? adapting pretrained representations to diverse tasks. In *Proc. of Association for Computational Linguistics (ACL)*.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.
- Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. 2021. Zero-shot text-to-image generation. *arXiv preprint arXiv:2102.12092*.
- Marc’Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. Sequence level training with recurrent neural networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*.
- Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Proc. of Neural Information Processing Systems (NIPS)*.
- Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. 2017. Self-critical sequence training for image captioning. In *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Sebastian Ruder. 2019. *Neural transfer learning for natural language processing*. Ph.D. thesis, NUI Galway.
- Abdelrhman Saleh, Natasha Jaques, Asma Ghandeharioun, Judy Shen, and Rosalind Picard. 2020. Hierarchical reinforcement learning for open-domain dialog. In *Proc. of Conference on Artificial Intelligence (AAAI)*.
- John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. 2016. High-dimensional continuous control using generalized advantage estimation. In *Proceedings of the International Conference on Learning Representations (ICLR)*.
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.
- Mathieu Seurin, Philippe Preux, and Olivier Pietquin. 2020. "i'm sorry dave, i'm afraid i can't do that" deep q-learning from forbidden actions. In *Proc. of the International Joint Conference on Neural Networks (IJCNN)*.
- David Silver, Satinder Singh, Doina Precup, and Richard S Sutton. 2021. Reward is enough. *Artificial Intelligence*, page 103535.
- Satinder Singh, Diane Litman, Michael Kearns, and Marilyn Walker. 2002. Optimizing dialogue management with reinforcement learning: Experiments with the njfun system. *Journal of Artificial Intelligence Research*, 16:105–133.
- Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. 2020. Learning to summarize from human feedback. *arXiv preprint arXiv:2009.01325*.

- Florian Strub, Harm De Vries, Jérémie Mary, Bilal Piot, Aaron Courville, and Olivier Pietquin. 2017. End-to-end optimization of goal-driven and visually grounded dialogue systems. In *Proc. of International Joint Conferences on Artificial Intelligence (IJCAI)*.
- Richard S Sutton, Andrew G Barto, et al. 1998. *Introduction to reinforcement learning*, volume 135. MIT press Cambridge.
- Richard S Sutton, David A McAllester, Satinder P Singh, Yishay Mansour, et al. 1999. Policy gradient methods for reinforcement learning with function approximation. In *Proc. of Neural Information Processing Systems (NIPS)*.
- Guy Tennenholtz and Shie Mannor. 2019. The natural language of actions. In *Proc. of International Conference on Machine Learning (ICML)*.
- Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Pei-Hsin Wang, Sheng-Iou Hsieh, Shih-Chieh Chang, Yu-Ting Chen, Jia-Yu Pan, Wei Wei, and Da-Chang Juan. 2020. Contextual temperature for language modeling. *arXiv preprint arXiv:2012.13575*.
- Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256.
- Qingyang Wu, Yichi Zhang, Yu Li, and Zhou Yu. 2019. Alternating roles dialog model with large-scale pre-trained language models. In *Proc. of European Association for Computational Linguistics (EACL)*.
- Jianwei Yang, Jiasen Lu, Stefan Lee, Dhruv Batra, and Devi Parikh. 2018. Visual curiosity: Learning to ask questions to learn visual recognition. In *Proc. of International Conference on Machine Learning (ICML)*.
- Shunyu Yao, Rohan Rao, Matthew Hausknecht, and Karthik Narasimhan. 2020. Keep calm and explore: Language models for action generation in text-based games. In *Proc. of Empirical Methods in Natural Language Processing (EMNLP)*.
- Tom Zahavy, Matan Haroush, Nadav Merlis, Daniel J Mankowitz, and Shie Mannor. 2018. Learn what not to learn: Action elimination with deep reinforcement learning. In *Proc. of Neural Information Processing Systems (NeurIPS)*.
- Hugh Zhang, Daniel Duckworth, Daphne Ippolito, and Arvind Neelakantan. 2020. Trading off diversity and quality in natural language generation. In *Proc. of European Association for Computational Linguistics (EACL)*.
- Xu Zhang, Felix Xinnan Yu, Svebor Karaman, Wei Zhang, and Shih-Fu Chang. 2018. Heated-up softmax embedding. *arXiv preprint arXiv:1809.04157*.
- Yizhe Zhang, Zhe Gan, Kai Fan, Zhi Chen, Ricardo Henao, Dinghan Shen, and Lawrence Carin. 2017. Adversarial feature matching for text generation. In *Proc. of International Conference on Machine Learning (ICML)*.
- Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*.

A Dataset and training details

A.1 Evaluation Metrics

For the BLEU and METEOR scores, we used the NLTK⁵ implementations with the smoothing function number 2 for the BLEU score. For the CIDEr score, we used the nlg-eval implementation⁶.

A.2 Answer filtering

For each dataset, we remove *yes* and *no* question-answer pairs which frequency largely exceeds other answers, to avoid any bias in the question generation process, as usually done in the VQG litterature ([Mostafazadeh et al., 2016](#)).

A.3 Dataset split

For CLEVR (resp. VQAv2), the RL language agent is trained for 50k (resp. 100k) episodes over the first 20k images (resp. all the images) of the training dataset, and is then evaluated on the first 5k (resp. 20k) images of the validation set. Besides, we uniformly sample the answer in the set of reference answers for each image to reduce the bias in the distribution of answers. Finally, questions are limited to 20 (resp. 10) words.

A.4 Language Agent Networks and Training

For CLEVR (resp. VQAv2), we used a single-layer LSTM with 64 (resp. 256) units for the policy network. At every time step, the LSTM input is then the concatenation of the word embedding of dimension 32 (resp. 128), the answer embedding of dimension 32 (resp. 128), and the image representation. For CLEVR, the image representation is extracted from a pretrained ResNet50 and projected into a tensor of size (32,7,7) before being flattened. For VQAv2, the image representation is the average of 200 bounding box features of dimension 1048, extracted from a faster R-CNN ([Ren et al., 2015](#)).

We optimize the full loss $L = L_{PPO} + \alpha L_{VF} + \beta L_E$ with $\alpha = 0.5$, $\beta = 0.01$ and a PPO clipping ratio $\epsilon = 0.02$ (resp. 0.01) for CLEVR (resp. VQAv2). We use Adam optimizer ([Kingma and Ba, 2014](#)) with a learning rate (lr) of 10^{-3} for TruflL and the scratch baseline, 10^{-5} (resp. 10^{-6}) for RL algorithms with a pre-training phase on CLEVR (resp. VQAv2), and $5*10^{-4}$ for models including a KL regularization term. We use a batch size (bs) of 128 for all models except the ones with KL regularization, for which we use a batch size of 64. Finally, for the RL from scratch baselines, we perform gradient clipping (gradclip) of 1 (resp. 5) for CLEVR and VQAv2.

Such hyper-parameters were selected, after conducting an extensive hyper-parameter search. The following values were tested: $\beta \in \{0.01, 0.02, 0.05, 0.1\}$, $\epsilon \in \{0.01, 0.02, 0.05, 0.1, 0.5, 0.9\}$, lr $\in \{10^{-6}, 10^{-5}, 10^{-4}, 5*10^{-4}, 10^{-3}, 5*10^{-3}, 10^{-2}, 5*10^{-2}\}$, gradclip $\in \{\text{None}, 1, 5, 10, 100\}$, bs $\in \{32, 64, 128\}$.

Additionally, we also tested for VQAv2 policy networks with 64, 256 and 1024 units, with respectively 32, 128 and 512 word embedding dimensions. We kept the network size giving the best performances, i.e. policy network of 256 units and 128 word embedding dimension.

A.5 Reward formula for VQAv2

In this section, we detail the reward function used for the VQAv2 task. $r(w_t, w_{<t}, c) = \mathbb{1}_{\text{rk}(\mathcal{A}) \leq 10, t=T-1} e^{-\text{rk}(\mathcal{A})/2}$, with $\text{rk}(\mathcal{A})$ the rank of the ground-truth answer given by the VQA model, when predicting the actual answer from the terminal state $(c, w_{<T})$. Formally, it is defined as:

$$\text{rk}(\mathcal{A}) = \text{rank}(\text{VQA}(c, w_{<T})[\mathcal{A}]),$$

with $\text{VQA}(c, w_{<T})$ the probability distribution given by the VQA model over the set of answers, and rank the function which ranks the probability of answer \mathcal{A} within $\text{VQA}(c, w_{<T})$ probability distribution.

B Additional experiments

B.1 CLEVR

Table 6 displays the complete ablation on the truncation functions with parameters sweep. The 'sizeVA' variable indicates the average size of the truncated action space for each truncation function. Table 7 displays the

⁵<https://www.nltk.org/>

⁶<https://github.com/Maluuba/nlg-eval>

ablation over the three decoding procedures defined in Section 4.5. Such an ablation presents a similar pattern than VQAv2 results described in section 5.2.

Finally, Table 8 reports CLEVR metrics when using the BLEU score as the reward. While on such a task TruFLL still exhibits promising language scores, the n-grams metrics remain lower than the pretrained baselines. This illustrates that using a language similarity score as a reward signal is much less interesting than a reward based on a task completion score.

Table 6: **CLEVR task:** Ablation on the truncation functions with parameters sweep. Best values are in bold.

trunc.	Score	BLEU	CIDEr	ppl-e(\downarrow)	sBLEU(\downarrow)	Size VA
TruFLL (Task-LM)						
top($k=10$)	0.50	0.12	0.32	10^2	0.93	10
top($k=20$)	0.45	0.10	0.24	10^3	0.87	20
$p_{th}(\alpha=0.05)$	0.55	0.18	0.63	25	0.96	4.4
$p_{th}(\alpha=0.1)$	0.47	0.18	0.87	6.7	0.98	2.4
$p_{th}(\alpha=1/V)$	0.50	0.16	0.49	41	0.97	6.6
top($p=0.85$)	0.52	0.17	0.69	10.4	0.96	4.6
top($p=0.9$)	0.51	0.17	0.69	11.5	0.96	5.1
sample($k=20$)	0.50	0.18	0.73	18.9	0.86	5.4
sample($k=30$)	0.50	0.18	0.73	16.1	0.89	6.1
TruFLL (Ext-LM)						
top($k=10$)	0.52	0.06	0.15	10^2	0.94	10
top($k=20$)	0.48	0.05	0.12	10^2	0.89	20
$p_{th}(\alpha=0.05)$	0.48	0.08	0.34	3.03	0.95	3.3
$p_{th}(\alpha=0.1)$	0.45	0.17	0.74	2.2	0.99	2.1
$p_{th}(\alpha=1/V)$	0.44	0.11	0.37	3.7	0.96	5.7
top($p=0.85$)	0.45	0.10	0.39	3.2	0.92	4.1
top($p=0.9$)	0.48	0.15	0.57	2.8	0.97	4.3
sample($k=20$)	0.45	0.14	0.50	2.4	0.92	4.1
sample($k=30$)	0.43	0.13	0.46	2.7	0.92	4.6

Table 7: **CLEVR task:** Ablation on sampling methods. Best overall values are underlined, while best values for TruFLL are in bold.

method	text-gen	score	BLEU	CIDEr	ppl-e
pretrain	greedy	0.32	<u>0.22</u>	<u>1.01</u>	14
	sampling	0.29	0.17	0.76	58
	lm-ranking	0.28	0.18	0.73	20
pretrain + RL	greedy	0.53	0.18	0.73	24
	sampling	0.40	0.16	0.68	39
	lm-ranking	0.40	0.17	0.68	5
Task-LM	greedy	0.57	0.17	0.65	39
	sampling	0.55	0.17	0.66	24
	lm-ranking	0.51	0.16	0.65	9
Ext-LM	greedy	0.48	0.09	$0.34(\pm 0.11)$	3.0
	sampling	0.48	0.10	$0.35(\pm 0.11)$	3.1
	lm-ranking	0.48	0.06	$0.34(\pm 0.11)$	<u>2.9</u>

B.2 VQAv2

Temperature scheduling: On the CLEVR task, we observed that dynamic truncations outperform static ones such as top(k): indeed, they better take into account the inherent variability of the language structure at the sentence-level. When scaling up to the 15k words of the VQAv2 task, we also dynamically decrease the truncation size through training, by applying a decreasing temperature schedule on the language model. While temperature scaling (Bahdanau et al., 2015) is usually used at test time to control the smoothness of the language model distribution, temperature schedules during training of language models have been used in several settings (Jang et al., 2016; Zhang et al., 2018; Wang et al., 2020). Formally, $f_{LM}(w_i|w_{<t})$ distribution is computed as $\text{softmax}(x_i) = e^{-x_i/\tau} / \sum_j e^{-x_j/\tau}$, with x_j the LM logits and τ the temperature, which decreases from τ_{max} to τ_{min} by a factor T_F every T_u training step. In Table 9, both TruFLL (Task-LM) and TruFLL

Table 8: **CLEVR, BLEU reward.** Scores are averaged over the three decoding procedures detailed in Section 4.5 and over 5 seeds, standard deviation are displayed whenever greater than 0.01 for accuracy metrics. We here report the models with the highest task-success, i.e. the *scratch with KL regularization* baseline with $\lambda_{KL}=0.1$, and the truncation model with a probability threshold, $p_{th}(\alpha=0.05)$. Baseline and Metrics are respectively detailed in Section 4.4 and 4.3. Best overall values are underlined, while best values for models without task-data (i.e RL from scratch algorithms) are in bold.

Method	Score	R@5	BLEU	Meteor	CIDEr	ppl-t (\downarrow)	ppl-e (\downarrow)	sBLEU (\downarrow)	peak. (\downarrow)
pretrain	0.30	0.71	0.19	0.38	0.83	3.1	31	0.44	0.96
pretrain + RL fine-tune	<u>0.34</u>	<u>0.80</u>	<u>0.20</u>	<u>0.38</u>	<u>0.83</u>	3.8	12	0.56	0.96
scratch	0.03	0.19	0.06	0.09	0.09	10^8	10^6	<u>0.13</u>	<u>0.14</u>
scratch + KL-task	0.09	0.33(± 0.15)	0.15	0.31	0.58(± 0.23)	3.8	63	0.34	0.95
scratch + KL-ext	0.06	0.30(± 0.23)	0.13	0.25	0.42	10^3	3.6	0.37	0.96
scratch + Truncation-task	0.17	0.51	0.18	0.37	0.80	<u>2.6</u>	17	0.63	1.0
scratch + Truncation-ext	0.07	0.36	0.16	0.29	0.49	10^2	<u>2.3</u>	0.60	1.0

(Ext-LM) benefit slightly from truncation with a temperature schedule compared to a vanilla truncation. The former displays the best performance/language scores trade-off for the schedule " $\tau: 3 > 1$. & $T_u=5,000$ ", while the latter has the best metrics trade-off for " $\tau: 1.5 > 1$. & $T_u=5,000$ ".

Finally, Figure 4 displays the evolution of the training return for TrufLL and the baselines. As expected, the pretrain+RL fine-tune baseline return does not evolve much, confirming that the policy distribution almost does not shift through the fine-tuning phase. The training curves of TrufLL present a steady increase in the return until reaching convergence, confirming that our approach, by guiding the exploration of the action space, provides a sufficient learning signal. On the other hand, the scratch+KL baselines stay stuck to a low training return. This suggests that the KL regularization term, while encouraging the policy distribution to resemble the language model distribution, fails to capture the task pragmatics, which requires generating a language that is visually grounded.

Table 9: **VQA task:** Ablation on the temperature schedules. "no temp. sch" is a classic truncation without temperature scheduling. We then report different schedules $\tau: \tau_{max} > \tau_{min}$, T_u , with τ_{max} , τ_{min} , T_u , and $T_f = 0.75$ as defined in section B.2. Best values are in bold.

Scheduling	Score	BLEU	CIDEr	ppl-e(\downarrow)	sBLEU(\downarrow)
TrufLL (Task-LM)					
no temp. sch	0.35	0.20	0.11	10^2	0.78
$\tau: 1.5 > 1$. $T_u=5,000$	0.34	0.18	0.11	10^2	0.79
$\tau: 3 > 1$. $T_u=5,000$	0.35	0.22	0.13	10^2	0.76
$\tau: 1.5 > 1$. $T_u=15,000$	0.31	0.23	0.23	10^2	0.73
TrufLL (Ext-LM)					
no temp. sch	0.34	0.18	0.04	25	0.83
$\tau: 1.5 > 1$. $T_u=5,000$	0.33	0.19	0.05	20	0.83
$\tau: 3 > 1$. $T_u=5,000$	0.32	0.15	0.05	35	0.82
$\tau: 1.5 > 1$. $T_u=15,000$	0.29	0.16	0.08	38	0.68

B.3 Additional discussion

TrufLL with a pre-training phase. Although TrufLL aims at providing a robust method to learn a language model (almost) from scratch, we investigate whether such algorithm can be complementary to RL algorithms with a pre-training phase. Therefore, when using the task-related dataset, we evaluate TrufLL from a pretrained policy, and we refer to it as TrufLL_{pretrain}.

In table 10, while on CLEVR, TrufLL_{pretrain} marginally improves the results of the pretrain+RL fine-tune baseline, the combination of TrufLL with a pre-training phase leads to performance degradation on VQAv2. This suggests that on a large vocabulary task, the language distribution learned by the SL pretrained policy is significantly different from the one learned with TrufLL.

On-policy TrufLL versus off-policy TrufLL. To ease off-policy learning, we propose to add a KL-regularization term in the RL loss (Jaques et al., 2017, 2019; Wu et al., 2019), and refer to it as TrufLL_{off,KL}. Intuitively, it encourages the policy to stay close to the language model's distribution, with a distribution support attributing negligible probabilities to words outside the truncated action space.

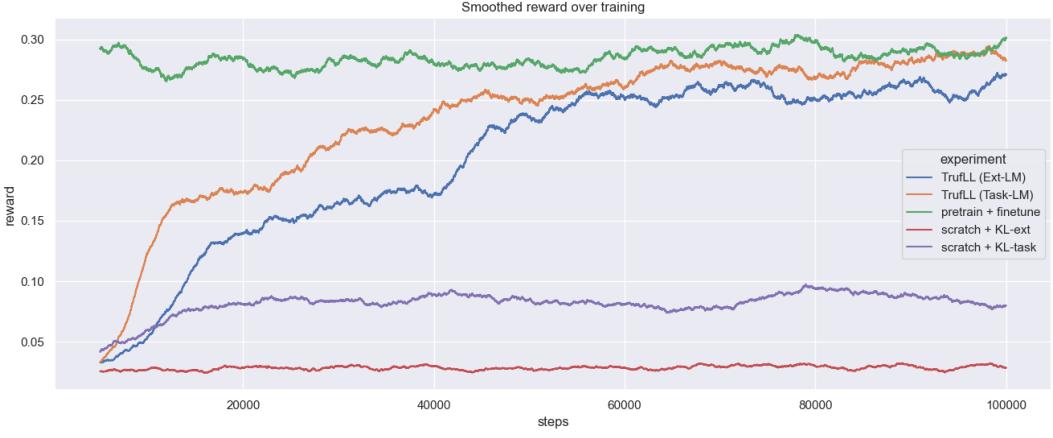


Figure 4: **VQAv2:** Training curves. Reward is a rolling average over 5000 timesteps.

Table 10: $\text{TruflL}_{\text{pretrain}}$ results on the 2 tasks. Additionally, we report the results for the pretrain+RL fine-tune baseline as a comparison. Best values are in bold.

Algo	Score	BLEU	CIDEr	ppl-e	sBLEU
CLEVR					
pretrain+RL	0.44	0.17	0.70	35	0.46
$\text{TruflL}_{\text{pretrain}}$	0.61	0.18	0.77	22	0.84
VQAv2					
pretrain+RL	0.41	0.31	0.98	50	0.78
$\text{TruflL}_{\text{pretrain}}$	0.33	0.27	0.42	35	1.0

Table 11 displays the full results of on-policy versus off-policy scores for TruflL (Task-LM) and TruflL (Ext-LM) on the two tasks. The full results emphasize the challenges of the approach for the large vocabulary of VQAv2. Indeed, on the off-policy setting for such a task, the exploding values for e-ppl suggest that the optimized language agent samples incoherent words taken outside the truncated action space, as corroborated by the low values of the sumVA ratio.

Interestingly, while on CLEVR, $\text{TruflL}_{\text{off,KL}}$ trades off task performance for language quality when compared to $\text{TruflL}_{\text{off}}$, on VQAv2, it mainly provides a better learning signal for the complete (large) vocabulary. In such a setting, it hence improves the global scores of the off-policy version of TruflL, and enables a much better generalization at test time of the global policy over the full vocabulary. Yet, keeping truncation at test time remains crucial with large vocabulary. Note that for VQAv2, the poor performances of $\text{TruflL}_{\text{off,KL}}$ on the external LM is mainly due to numerical instability challenges when using GPT-2 as the target policy of the KL regularization term.

Additionally, on-policy versus off-policy scores split per sampling procedure are displayed in table 12: unsurprisingly, greedy decoding for $\text{TruflL}_{\text{off}}$ outperforms the two sampling-based methods, that are more penalized by the imperfect generalization of the optimized policy over the full vocabulary.

Table 11: On-policy vs. off-policy scores for different variants of TrufLL: when training with an off-policy loss, we remove the truncation at test time. $\text{TrufLL}_{\text{off},\text{KL}}$ is evaluated with $\lambda_{KL} = 0.05$. Best values are in bold.

Algo	Score	BLEU	CIDEr ppl-e	sBLEU	sumVA
CLEVR					
TrufLL (Task-LM)					
TrufLL	0.56	0.17	0.06 10^3	0.78	N.A
TrufLL _{off}	0.50	0.14	0.43 10^4	0.88	0.96
TrufLL _{off,KL}	0.39	0.17	0.71 69	0.48	0.95
TrufLL (Ext-LM)					
TrufLL	0.48	0.08	0.34 3.03	0.95	N.A
TrufLL _{off}	0.41	0.10	0.35 10^5	0.88	0.95
TrufLL _{off,KL}	0.35	0.15	0.60 20	0.55	0.96
VQAv2					
TrufLL (Task-LM)					
TrufLL	0.35	0.21	0.11 10^4	0.36	N.A
TrufLL _{off}	0.07	0.03	0.01 10^4	0.05	0.08
TrufLL _{off,KL}	0.12	0.24	0.25 10^3	0.26	0.71
TrufLL (Ext-LM)					
TrufLL	0.34	0.18	0.04 24	0.83	N.A
TrufLL _{off}	0.09	0.04	0.01 10^4	0.05	0.07
TrufLL _{off,KL}	0.0	0.15	0.02 10^3	0.19	0.47

Table 12: On-policy vs. off-policy scores per decoding procedure: when training with an off-policy loss, we remove the truncation at test time. $\text{TrufLL}_{\text{off}, \text{KL}}$ is evaluated with $\lambda_{KL} = 0.05$. Best values are in bold.

method	text-gen	score	BLEU	CIDEr	e-ppl
CLEVR					
TrufLL (Task-LM)					
	greedy	0.57	0.17	0.65	39
TrufLL	sampling	0.55	0.17	0.66	24
	lm-ranking	0.51	0.16	0.65	8.8
	greedy	0.52	0.17	0.58	71
TrufLL _{off}	sampling	0.49	0.16	0.59	10^5
	lm-ranking	0.48	0.17	0.58	19
	greedy	0.56	0.18	0.78	24
TrufLL _{off, KL}	sampling	0.31	0.16	0.62	10^2
	lm-ranking	0.31	0.18	0.74	5.8
TrufLL (Ext-LM)					
	greedy	0.48	0.09	0.34	3.1
TrufLL	sampling	0.48	0.10	0.35	3.1
	lm-ranking	0.48	0.06	0.34	2.9
	greedy	0.42	0.10	0.38	4.4
TrufLL _{off}	sampling	0.40	0.10	0.35	10^6
	lm-ranking	0.40	0.10	0.34	15
	greedy	0.48	0.16	0.70	2.1
TrufLL _{off, KL}	sampling	0.27	0.13	0.48	55
	lm-ranking	0.30	0.16	0.61	2.0
VQAv2					
TrufLL (Task-LM)					
	greedy	0.36	0.20	0.11	366
TrufLL	sampling	0.35	0.20	0.11	337
	lm-ranking	0.34	0.21	0.11	95
	greedy	0.09	0.04	0.02	10^3
TrufLL _{off}	sampling	0.05	0.03	0.01	10^6
	lm-ranking	0.06	0.03	0.01	10^4
	greedy	0.16	0.29	0.46	38
TrufLL _{off, KL}	sampling	0.08	0.19	0.09	10^4
	lm-ranking	0.12	0.24	0.22	10^2
TrufLL (Ext-LM)					
	greedy	0.48	0.09	0.34	3.1
TrufLL	sampling	0.48	0.10	0.35	3.1
	lm-ranking	0.48	0.06	0.34	2.9
	greedy	0.11	0.05	0.01	10^2
TrufLL _{off}	sampling	0.07	0.03	0.01	10^5
	lm-ranking	0.08	0.04	0.01	10^4
	greedy	0.00	0.18	0.05	27
TrufLL _{off, KL}	sampling	0.00	0.13	0.01	10^3
	lm-ranking	0.00	0.16	0.02	10^2

C Human Evaluation details

For the Human Evaluation study, we designed one form per participant, with three sections evaluating respectively the language quality, language grounding and diversity criteria. Given the five evaluated models, there are ten different model pairs: each section of the form contains 10 pairwise comparison covering all the possible model pairs for the criteria. Each pairwise comparison is sampled uniformly over the 50 first question samples generated by the algorithms at test time. The evaluation of syntax errors was made within the diversity section: for each questions pair, we asked participants to tick the questions if they are grammatically incorrect. Figure 5 displays one pairwise comparison example for the three sections, and a full form example is available at the following url: <https://forms.gle/kkL38x31wF7A9YKx5>.

Evaluation of question quality

Select the question that is the most syntactically and semantically correct, and is more likely to be human language.

Pair #4 *

Q1: What is the cat inside of?

Q2: What color is the man's shirt?

(a) Language Quality pairwise comparison

Question Relevancy with image and answer

Select the question that is the most suitable given the image and answer.

Answer: california



Pair #2 *

Q1: What is the girl wearing on his shirt?

Q2: Where could it must be appropriate found?

(b) Language Grounding pairwise comparison

Figure 5: Examples of pairwise comparison for each evaluated criteria.

Question Diversity with a reference question

Select the question that is the most different from the reference question (or the most original).

Reference question: What color spot does the horse have? *

Q1: What color is the animal?

Q2: Which color is his socks?

Tick the question if it is grammatically incorrect.

Q1: What color is the animal?

Q2: Which color is his socks?

(c) Diversity/Originality with reference question. Pairwise comparison and evaluation of syntax errors.

Figure 5: Examples of pairwise comparison for each evaluated criteria. (cont.)

D Additional VQG Samples

Figure 6 and Figure 7 display the 10 first dialog samples produced at test time on CLEVR, while figures 8, 9, and 10 display the 15 first dialog samples produced at test time on VQAv2.

	Human	the big yellow object is what shape ? A:sphere
	pretrain pretrain + RL	there is a small thing that is the same color as the small rubber cylinder ; what is its shape? What is on the person's head?
	scratch scratch+KL-task scratch+KL-ext	yellow on or an material ? what number of other things are the same shape as the small gray thing ? does that tiny object have objects to its left ?
	TruFL (Task-LM) TruFL (Ext-LM)	what shape is the big thing that is to the right of the big matte thing ? what is the shape of the big object that is behind the big yellow thing and is the same color?
	Human	what number of other objects are the same size as the red rubber thing ? A:3
	pretrain pretrain + RL	how many other things are there of the same size as the purple rubber thing ? what number of objects are either large green metallic objects or tiny objects that are behind the tiny purple metallic thing?
	scratch scratch+KL-task scratch+KL-ext	many tiny cubes things things things both things as things that what number of other objects are there of the same material as the tiny cyan thing ? are there any blue objects ?
	TruFL (Task-LM) TruFL (Ext-LM)	what number of objects are either big objects in front of the small yellow object or big matte objects? how many objects in front of the big object ?
	Human	what number of other things are there of the same material as the large green object? A:3
	pretrain pretrain + RL	how many other things are there of the same size as the purple rubber cylinder ? what number of objects are either tiny cyan things or big cyan things ?
	scratch scratch+KL-task scratch+KL-ext	many tiny cubes things things things both things as things that what number of other objects are the same shape as the small yellow object ? how many things does that large thing have to its behind ?
	TruFL (Task-LM) TruFL (Ext-LM)	what number of other things are there of the same size as the green cylinder ? how many objects in front of the in the cylinder ?
	Human	what number of other things are there of the same shape as the small purple metallic thing ? A:1
	pretrain pretrain + RL	what number of other objects are the same color as the tiny rubber cylinder ? what number of purple objects are either small matte objects or big matte blocks ?
	scratch scratch+KL-task scratch+KL-ext	many gray in big purple purple purple many or many gray matte matte what number of other things are the same color as the large rubber cylinder ? how many other things in the are of same color as the large cylinder ?
	TruFL (Task-LM) TruFL (Ext-LM)	how many tiny things have the same color as the large rubber thing ? how many other things in the are of the same color as that large thing ?
	Human	what shape is the big matte object that is on the right side of the big cyan matte object ? A:cylinder
	pretrain pretrain + RL	the cyan matte thing that is the same size as the brown object is what shape ? what shape is the cyan matte object that is behind the cylinder ?
	scratch scratch+KL-task scratch+KL-ext	many yellow big either either that that more that metal ? what number of other things are the same shape as the small gray thing ? what number of blocks are in the things in the ?
	TruFL (Task-LM) TruFL (Ext-LM)	how many tiny things have the same color as the large rubber thing ? what is the shape of that large thing ?

Figure 6: Samples on CLEVR.

	Human	what is the size of the other rubber cylinder that is the same color as the big cylinder ? A:small
	pretrain pretrain + RL	there is a purple object that is the same size as the purple rubber cylinder ; what is its shape? what size is the gray ball that is right of the purple sphere ?
	scratch scratch+KL-task scratch+KL-ext	that greater tiny as shiny both are a tiny it either ball right there is a big thing that is the same color as the big matte cylinder ; what is its shape? how material is the yellow ?
	TruFL (Task-LM) TruFL (Ext-LM)	how big is the thing that is to the right of the big matte thing ? what size is the object that is behind the large red thing ?
	Human	There is a shiny thing that is both right of the small matte thing and behind the large yellow cube; what size is it ? A:small
	pretrain pretrain + RL	there is a big thing that is the same color as the small rubber cylinder ; what is its shape there is a brown matte object to the right of the cyan object ; what shape is it ?
	scratch scratch+KL-task scratch+KL-ext	many yellow big either either that that more that metal ? what number of other things are the same shape as the small gray thing ? what is the material of that block ?
	TruFL (Task-LM) TruFL (Ext-LM)	what shape is the big thing that is to the right of the big cyan thing ? what is the shape of that large thing ?
	Human	there is a object that is the same color as the rubber cylinder ;what is its shape ? A:sphere
	pretrain pretrain + RL	there is a small thing that is the same color as the small rubber cylinder ; what is its shape? what shape is that is the same color as the cylinder ?
	scratch scratch+KL-task scratch+KL-ext	many yellow yellow and cube shape behind cubes shape less small equal shape small equal large large ? how many other things in the color are of same material as the green shiny object ? how many spheres anything ?
	TruFL (Task-LM) TruFL (Ext-LM)	what is the shape of the small cyan thing ? And shape ?
	Human	what is the color of the small thing that is the same shape as the large gray object ? A:green
	pretrain pretrain + RL	there is another rubber object that is the same shape as the small brown object ; what color is it? what is the color of the tiny rubber thing ?
	scratch scratch+KL-task scratch+KL-ext	many sphere less how an what number of other objects are the same shape as the tiny blue object ? what size is that cylinder ?
	TruFL (Task-LM) TruFL (Ext-LM)	what is the color of the tiny matte thing ? what color is the small thing ?
	Human	what number of shiny objects are cyan spheres or tiny balls ? A:4
	pretrain pretrain + RL	how many other things are there of the same size as the brown rubber thing ? how many other things are there of the same size as the cyan rubber thing ?
	scratch scratch+KL-task scratch+KL-ext	many yellow do do either do either do balls what number of other things are there of the same shape as the small gray thing ? how many other things are of same color as ball ?
	TruFL (Task-LM) TruFL (Ext-LM)	how many other things are the same material as the small cyan cylinder ? how many other things in the material of the small thing that is the same material as green thing ?

Figure 7: Samples on Clevr.



Human	How many trains? A:1
pretrain	How many trains are in the picture?
pretrain + RL	How many trains are shown?
scratch	Amazing Amazing Amazing Amazing Amazing Amazing
scratch+KL-task	How many people are in the picture?
scratch+KL-ext	The same way of the same way of the
TruFL (Task-LM)	How many windows are here?
TruFL (Ext-LM)	How many different ways would we take them to reach



Human	What is the man wearing over his shirt? A:nothing
pretrain	What is in front of the person?
pretrain + RL	What is written on the person's right?
scratch	Amazing Amazing Amazing Amazing Amazing
scratch+KL-task	Is this child old or unloaded?
scratch+KL-ext	The first thing of the world's a few
TruFL (Task-LM)	How many different shapes have been used and used by
TruFL (Ext-LM)	How many planes can be seen on top?



Human	Is the fence in the background metal or wood? A:metal
pretrain	What kind of dog is on the dog?
pretrain + RL	What kind of dog is this?
scratch	Amazing Amazing Amazing Amazing Amazing
scratch+KL-task	What is 150 seated over the where?
scratch+KL-ext	The same way of the same way to be
TruFL (Task-LM)	Which item has more aggressive up their legs?
TruFL (Ext-LM)	How many different species have been captured and tagged here



Human	What is the title of the red book? A:harry potter
pretrain	What is the cat inside of?
pretrain + RL	Is the cat inside or outside?
scratch	Amazing Amazing Amazing Amazing Amazing
scratch+KL-task	What color is the man's shirt?
scratch+KL-ext	The way of the world's a lot of the
TruFL (Task-LM)	Which item appears higher into one and lower?
TruFL (Ext-LM)	What was that?



Human	What color is the fire hydrant? A:red
pretrain	What color is the hydrant?
pretrain + RL	What color is the hydrant?
scratch	Amazing Amazing Amazing Amazing Amazing
scratch+KL-task	What color is the man's bat?
scratch+KL-ext	The first thing is a good thing that the
TruFL (Task-LM)	Which color is this fire?
TruFL (Ext-LM)	What color will your feet color look?



Human	How many wheels does the truck have? A:3
pretrain	How many people are in front of the bus?
pretrain + RL	How many slices ofists are on the plate?
scratch	Amazing Amazing Amazing Amazing Amazing
scratch+KL-task	Is summer out or cloudy next to Winchester?
scratch+KL-ext	The most recent of the most recent years of
TruFL (Task-LM)	How many pieces are here?
TruFL (Ext-LM)	How many different objects have been used?

Figure 8: Samples on VQA.



Human	What is on top of the round dome? A:flag
pretrain	What is on the top right mean?
pretrain + RL	What is on the front of this event?
scratch	Amazing Amazing Amazing Amazing Amazing
scratch+KL-task	What zombie is on the mouse?
scratch+KL-ext	The the first thing is a bit of the first
TruflL (Task-LM)	Where could one travel park located?
TruflL (Ext-LM)	What color will your shoes look?



Human	Where is the chain link fence? A:background
pretrain	What is he holding?
pretrain + RL	What is he fire hydrant?
scratch	mazing Amazing Amazing Amazing Amazing
scratch+KL-task	Who is closest to the paint?
scratch+KL-ext	The the first thing is a great deal with the
TruflL (Task-LM)	Which item represents than both ends and lower?
TruflL (Ext-LM)	How much food has it given him?



Human	What activity are these people doing? A:surfing
pretrain	What is the person doing?
pretrain + RL	What is the person doing?
scratch	noodles noodles noodles noodles noodles
scratch+KL-task	How many umbrellas are visible?
scratch+KL-ext	The the first thing is the same way of the
TruflL (Task-LM)	Which game does he play?
TruflL (Ext-LM)	What was that for?



Human	What color is the umbrella? A:black
pretrain	What color is the cat?
pretrain + RL	What color is the cat?
scratch	Amazing Amazing Amazing Amazing Amazing
scratch+KL-task	What color is the man's shirt?
scratch+KL-ext	The the other way of the past time, and
TruflL (Task-LM)	Which item doesn't both turn?
TruflL (Ext-LM)	What color of clothing did he get?



Human	How many planes are shown? A:1
pretrain	How many jets are there?
pretrain + RL	How many jets are there?
scratch	Amazing Amazing Amazing Amazing Amazing
scratch+KL-task	How many skater does Green cents have?
scratch+KL-ext	The the first thing is the first time, and
TruflL (Task-LM)	How many surf worthy are here?
TruflL (Ext-LM)	How many different ways should one ask if she wants



Human	What is this animal called? A:horse
pretrain	What is the animal on?
pretrain + RL	What animal is shown on the ground?
scratch	Amazing Amazing Amazing Amazing Amazing
scratch+KL-task	What has to make of the pies that, should
scratch+KL-ext	The the next week of the next week, the
TruflL (Task-LM)	Which item doesn't turn?
TruflL (Ext-LM)	What was that?

Figure 9: Samples on VQA.



Human	What color spot does the horse have? A:white
pretrain	What color is the animal?
pretrain + RL	What color is the door?
scratch	Amazing
scratch+KL-task	Amazing
scratch+KL-ext	Amazing
TruflL (Task-LM)	What color is the ATM basketball?
TruflL (Ext-LM)	The same thing that the same way of the
	Which color is his socks?
	What color will your shoes look?



Human	What color is the girls pants? A:blue
pretrain	What color is the man's blue?
pretrain + RL	What color are the bird's pants?
scratch	Amazing
scratch+KL-task	Amazing
scratch+KL-ext	Amazing
TruflL (Task-LM)	Amazing
TruflL (Ext-LM)	Amazing
	The first thing is a lot of the same
	Which color is this fire?
	What color of clothing did he get?



Human	What is on the woman's head? A:helmet
pretrain	What is on the girl's head?
pretrain + RL	What is on the person's head?
scratch	Amazing
scratch+KL-task	Amazing
scratch+KL-ext	Amazing
TruflL (Task-LM)	Who is behind the horse?
TruflL (Ext-LM)	The same thing that the most important to the
	Which item doesn't turn?
	What was that?

Figure 10: Samples on VQA.