Thinking Outside the (Gray) Box: A Context-Based Score for Assessing Value and Originality in Neural Text Generation

Anonymous ACL submission

Abstract

Despite the increasing use of large language models for creative tasks, their outputs often lack diversity. Common solutions, such as sampling at higher temperatures, can compromise the quality of the results. Dealing with this trade-off is still an open challenge in designing AI systems for creativity. Drawing on information theory, we propose a context-based score to quantitatively evaluate value and originality. This score incentivizes accuracy and adherence to the request while fostering divergence from the learned distribution. We show that our score can be used as a reward in a reinforcement learning framework to fine-tune large language models for maximum performance. We validate our strategy through experiments considering a variety of creative tasks, such as poetry generation and math problem solving, demonstrating that it enhances the value and originality of the generated solutions.

1 Introduction

Foundation models, particularly large language models (LLMs), are significantly transforming creative activities. They can serve as a foundation for co-creation systems involving human and artificial authors (Lin et al., 2023), can generate software code (Rozière et al., 2023), or even foster scientific research (Si et al., 2025). However, the nature of the self-supervised learning algorithms used for the training of these models tends to make their sampling distribution as close as possible to the training data distribution (Franceschelli and Musolesi, 2024a). In addition, fine-tuning, such as that based on reinforcement learning from human feedback (RLHF) (Christiano et al., 2017), is often necessary to generate appropriate and accurate responses. However, this process tends to reduce output diversity further (Kirk et al., 2024), and linguistic creativity tends to be lower than that of humans (Lu et al., 2025). On the contrary, LLMs for creative tasks should produce more novel and surprising texts that maintain a high level of correctness and adherence to the request. A common solution is to sample at a higher temperature to increase diversity. However, this might lead to generating less coherent text (Peeperkorn et al., 2024).

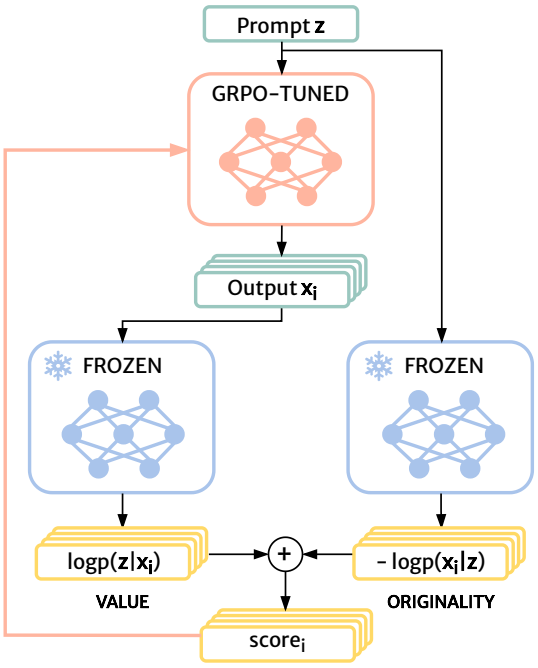

Figure 1: A summary of our method: the target model (orange) produces G outputs for each prompt; a frozen reference model (blue) computes the value and originality for each output; the overall scores are used in GRPO to correct the target model (orange line).

To address the issues described above, we propose a new training approach for creative tasks based on **CoVO**, a **Co**ntext-based score for **V**alue and **O**riginality, to take into consideration both value and originality, i.e., two core dimensions of creativity (Boden, 2003; Lubart, 1999; Runco and Jaeger, 2012), of the generated text in the optimization of LLMs. The definition of CoVO is grounded in the analysis of mutual information (MacKay,

2003) between the model's outputs and inputs, and vice versa. More specifically, we formulate a new optimization objective where, given a specific input, the desired output is derived by *simultaneously* maximizing the conditional probability of the input given the output and minimizing the conditional probability of the output given the input under the generative model. In this way, we optimize for solutions that are appropriate for the input request but also different from the outputs we would normally obtain from the model. In particular, we show that our information-theoretic score can be used as a reward in RL-based fine-tuning algorithms such as Group Relative Policy Optimization (GRPO) (Shao et al., 2024), guiding pre-trained and instructed models toward more diverse yet valuable solutions (see Figure 1 for a schematic overview). Evaluations on mathematical problem solving, poetry generation, and the tasks included in *NoveltyBench* (Zhang et al., 2025) demonstrate that our approach can enhance the quality and diversity of generated outputs, positioning it as a strong candidate for creativity-focused applications of current foundation models.

2 Related Work

2.1 Information Theory and Creativity

The quest to provide a mathematical and computational definition of creativity has been a significant focus in recent decades. Numerous methods have been developed to define various dimensions or attributes for evaluating the creativity of AI-generated products (see, for example, (Franceschelli and Musolesi, 2024a)). However, these methods are often domain-specific and typically require substantial human effort to implement and assess. In contrast, solutions based on information theory (Shannon, 1948; Cover, 1999) offer a more universally applicable approach.

Information-theoretic methods can quantify creativity by measuring the novelty and complexity of generated outputs, without the need for extensive human intervention, making them suitable for a wide range of domains. Bayesian surprise (Baldi and Itti, 2010), i.e., the divergence between a prior and a posterior belief, has been extensively used to measure different shades of originality, such as novelty and surprise (Lamb et al., 2019). Nevertheless, Varshney (2019) demonstrates that there is a mathematical limit for Bayesian surprise when combined with quality measures. Surprisal (Tribus, 1961), i.e., Shannon's self-information, has also been used (Bunescu and Uduehi, 2019); Barto et al. (2013) extensively discuss surprisal, Bayesian surprise, and novelty. Crucially, in the context of reinforcement learning (RL), surprisal has been used as a form of intrinsic motivation to encourage the agent to explore more (Achiam and Sastry, 2016). Sun et al. (2025) apply this idea to improve exploration in RLHF (Christiano et al., 2017). A similar strategy can be applied during LLM fine-tuning, either by explicitly maximizing the model's entropy (Cheng et al., 2026) or by decoupling entropy and cross-entropy from KL regularization and assigning greater weight to the former (Slocum et al., 2025). Additionally, mutual information has been applied to neural conversation models to improve both diversity and appropriateness (Li et al., 2016). However, all these existing approaches are not able to capture and simultaneously optimize value and originality at the same time.

2.2 LLMs and Creativity

Since the explosion of interest in LLMs in recent years, researchers have been keenly exploring their potential to exhibit creativity and ways for enhancing it (Franceschelli and Musolesi, 2025; Vadivel, 2026). From an evaluation standpoint, research has examined whether human creativity tests can be adapted to assess the creative capabilities of LLMs (Stevenson et al., 2022), which evaluation frameworks are most appropriate (Lu et al., 2026; Saakyan et al., 2026; Shaib et al., 2025), whether AI-generated texts can rival human-authored ones (Porter and Machery, 2024; Davis, 2024), and how these models may influence human creative processes (Doshi and Hauser, 2024; Lee and Chung, 2024). From a generation standpoint, creativity has been addressed either at inference time or during post-training. At inference time, several decoding strategies have been proposed to improve output diversity (e.g., (Minh et al., 2025; Potraghloo et al., 2025; Troshin et al., 2025)); prompting techniques, such as multi-perspective brainstorming (Lagzian et al., 2025) or zero-shot prompting to emulate the writing styles of famous authors (Sawicki et al., 2023), have also been explored. However, these techniques still depend on the acquired skills and are therefore constrained by the limited diversity inherent in them. Post-training approaches, instead, aim to steer the model toward richer and more diverse regions of the solution space. This can be performed through the application of RLHF (Chris-

tiano et al., 2017), for example to write short poems that human evaluators find more creative (Pardinas et al., 2023), or through RL algorithm variants (Anschel et al., 2025; Chung et al., 2025) and additional intrinsic rewards (Chen et al., 2025; Dai et al., 2026; Li et al., 2025), e.g., to promote valid but under-represented solutions (He et al., 2025), better balance exploration and exploitation (Cheng et al., 2026), or optimize metrics commonly linked to creative expression (Ismayilzada et al., 2025).

3 Preliminaries

3.1 Language Modeling

A $\boldsymbol{\theta}$-parameterized autoregressive language model is a probability distribution $p_{\boldsymbol{\theta}}(\mathbf{x})$ over a variable-length text sequence $\mathbf{x} = (x_1 \dots x_T)$, where T is the sequence length and each token x_t is in a finite vocabulary $\mathcal{V}$ of size N. The probability distribution is factorized as $p_{\boldsymbol{\theta}}(\mathbf{x}) = \prod_{t=1}^{T} p_{\boldsymbol{\theta}}(x_t|\mathbf{x_{<t}})$, where $\mathbf{x_{<t}} = x_1 \dots x_{t-1}$. The language model is usually trained to maximize the likelihood of the true distribution $p^*(\mathbf{x})$ for any $\mathbf{x}$ from a reference dataset (the training set). In other words, given an input $\mathbf{x_{<t}}$, the model learns to approximate the probability of each token from $\mathcal{V}$ being x_t. While this makes such a model immediately capable of scoring the probability of a given text, it also allows for the generation of new sentences. Given a conditional input (prompt) $\mathbf{z} = (z_1 \dots z_L)$, we can decode $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$ as the continuation of $\mathbf{z}$, i.e., through the factorized representation $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}) = \prod_{t=1}^{T} p_{\boldsymbol{\theta}}(x_t|\mathbf{x_{<t}}, \mathbf{z})$.

3.2 Reinforcement Learning for Language Models

Due to its adherence to the formal framework of Markov decision processes, RL can be used as a solution to the generative modeling problem in the case of autoregressive tasks such as text generation. The LLM plays the role of the agent, and each generated token represents an action a_t. The current version of the generated output $\mathbf{x_t}$ is part of the state $\mathbf{s_t}$ (potentially with additional information such as initial prompts). Finally, the reward r_{t+1} measures the "quality" of the current output. A common strategy is to assign a zero reward for each $\mathbf{x}_t, t \neq T$ and a sentence-based reward when the final output is generated. Within this framework, any policy-based method can be employed to train or fine-tune the LLM to optimize a given objective. Indeed, RL facilitates the use of non-differentiable reward functions, enabling the optimization of test-time metrics, domain-specific targets, and human preferences (Franceschelli and Musolesi, 2024b).

A widely used RL algorithm for fine-tuning LLMs is Proximal Policy Optimization (Schulman et al., 2017), which aims to maximize the following objective:

$$\mathcal{J}(\boldsymbol{\theta}) = \hat{\mathbb{E}}_t \Big[\min(r_t(\boldsymbol{\theta})\hat{A}_t, \texttt{clip}(r_t(\boldsymbol{\theta}), 1-\epsilon, 1+\epsilon)\hat{A}_t) \Big] \tag{1}$$

where $r_t(\boldsymbol{\theta}) = \frac{\pi_{\boldsymbol{\theta}}(\mathbf{x}_t|\mathbf{z},\mathbf{x}_{<t})}{\pi_{\boldsymbol{\theta}_{old}}(\mathbf{x}_t|\mathbf{z},\mathbf{x}_{<t})}$ with $\pi_{\boldsymbol{\theta}}$ and $\pi_{\boldsymbol{\theta}_{old}}$ denoting the current and old policy models, repectively; $\mathbf{x}$ is the output sampled from the old policy given the prompt $\mathbf{z}$; and ϵ is a clipping factor used to stabilize training. The advantage $\hat{A}_t$ is usually computed through Generalized Advantage Estimation (Schulman et al., 2016) based on the full rewards:

$$R(\mathbf{z}, \mathbf{x}) = r(\mathbf{z}, \mathbf{x}) - \beta \log \frac{\pi_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})}{\pi_{ref}(\mathbf{x}|\mathbf{z})}, \tag{2}$$

thus integrating a KL penalty with respect to a reference model (usually, the same model before fine-tuning), and a learned value function $v_{\boldsymbol{\phi}}(\mathbf{s})$. However, learning such a value function is computationally intensive, and its training is complicated by the fact that only the last state is scored by the reward function. Moreover, the inclusion of the KL penalty as an auxiliary reward term adds complexity to the advantage estimation process. To address these issues, Group Relative Policy Optimization (GRPO) (Shao et al., 2024) has been introduced. GRPO directly adds a KL divergence term $-\beta D_{KL}(\pi_{\boldsymbol{\theta}}||\pi_{ref})$ to the loss rather than to the single rewards, and especially obviates the need for a value function approximator by using the average reward of multiple sampled outputs as the baseline:

$$\hat{A}_{i,t} = \frac{r_i - \texttt{mean}(\mathbf{r})}{\texttt{std}(\mathbf{r})}, \tag{3}$$

with $\mathbf{r} = \{r_1, r_2, ..., r_G\}$ as the list of rewards received by each of the G outputs sampled from the same prompt.

4 A Context-Based Score for Valuable and Original Generation

Our goal is to derive a score that is able to quantify both value and originality at the same time. As discussed in depth by Csikszentmihalyi (2014), creativity depends on the context in which the product

is created, as the context provides the task identification and the domain information necessary to generate and validate the outcome. In turn, the output aims to solve the given task and provide a meaningful, original contribution to the current domain. Thus, our proposed score has its roots in mutual information, which represents a quantitative way to study the relationship between contextual, prior information, and a produced posterior outcome. More specifically, we start from the (pointwise) mutual information between two variables a and b:

$$I(a,b) = h(a) - h(a|b) = h(b) - h(b|a) \quad (4)$$

where the self-information is $h(y) = -\log p(y)$:

$$I(a,b) = \log p(a|b) - \log p(a) = \log p(b|a) - \log p(b). \quad (5)$$

Let us now assume a to be our input vector $\mathbf{z}$ and b our output vector $\mathbf{x}$, obtaining:

$$I(\mathbf{z},\mathbf{x}) = \log p(\mathbf{x}|\mathbf{z}) - \log p(\mathbf{x}). \quad (6)$$

We can generalize $I(\mathbf{z},\mathbf{x})$ with two scaling factors:

$$I(\mathbf{z},\mathbf{x},\lambda_1,\lambda_2) = \lambda_1 \log p(\mathbf{x}|\mathbf{z}) - \lambda_2 \log p(\mathbf{x}), \quad (7)$$

where $I(\mathbf{z},\mathbf{x})$ is just $I(\mathbf{z},\mathbf{x},1,1)$. Computing the *absolute* probability $p(\mathbf{x})$ can be difficult, as usually generative models are developed to assign *conditional* probabilities. By applying the Bayes theorem, i.e., $\log p(a|b) = \log p(b|a) + \log p(a) - \log p(b)$, we can substitute the $\log p(\mathbf{x})$ term as follows:

$$\begin{aligned} I(\mathbf{z},\mathbf{x},\lambda_1,\lambda_2) &= \lambda_1 \log p(\mathbf{x}|\mathbf{z}) - \lambda_2 \log p(\mathbf{x}|\mathbf{z}) \\ &\quad - \lambda_2 \log p(\mathbf{z}) + \lambda_2 \log p(\mathbf{z}|\mathbf{x}) \\ &= (\lambda_1 - \lambda_2) \log p(\mathbf{x}|\mathbf{z}) + \lambda_2 \log p(\mathbf{z}|\mathbf{x}) - \lambda_2 \log p(\mathbf{z}). \end{aligned} \quad (8)$$

Since our goal is to find the optimal $\mathbf{x}$ for a given $\mathbf{z}$, the last term can be ignored. Moreover, we now define $\lambda_v = \lambda_2$ and $\lambda_o = \lambda_2 - \lambda_1$, thus obtaining the following objective:

$$\overline{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmax}} (\lambda_v \log p(\mathbf{z}|\mathbf{x}) - \lambda_o \log p(\mathbf{x}|\mathbf{z})). \quad (9)$$

Let us now consider the case where $\lambda_v, \lambda_o > 0$, for example, $\lambda_v = \lambda_o = 1$. Solving this maximization problem involves finding the target $\mathbf{x}$ that maximizes the posterior probability of $\mathbf{z}$ while also being unlikely given $\mathbf{z}$. In other words, the optimal $\mathbf{x}^*$ must be unexpected and diverse from $p(\mathbf{x}|\mathbf{z})$, but it must also make $\mathbf{z}$ deducible from $\mathbf{x}^*$. $-\log p(\mathbf{x}|\mathbf{z})$, commonly known as surprisal (Tribus, 1961), is widely used to measure diversity and surprise (Barto et al., 2013), and adheres to the first requirement from the standard definition of creativity by Runco and Jaeger (2012), i.e., *originality* (where "original" is defined as "unusual"). Conversely, $\log p(\mathbf{z}|\mathbf{x})$ may be interpreted as reflecting the degree to which $\mathbf{x}$ effectively addresses $\mathbf{z}$, aligning with the second requirement of the definition and a property central to creative *value*. If the request (e.g., a problem or task) can be inferred from the outcome, the latter constitutes an appropriate instance of that task or a correct, useful solution to that problem (e.g., if the request is for a sonnet or a sci-fi screenplay, the generated artifact is valuable if identified as a poem satisfying the metrical constraints of a sonnet or as a screenplay adhering to a sci-fi theme, i.e., if it maximizes the posterior probability of being recognized as a sonnet or a sci-fi screenplay).

In summary, the **CoVO** (**Co**ntext-based **V**alue and **O**riginality) score for a target $\mathbf{x}$ given a source $\mathbf{z}$ on a reference probability distribution p is formally defined as:

$$s_{CoVO}(\mathbf{z},\mathbf{x},p) = \underbrace{\lambda_v \log p(\mathbf{z}|\mathbf{x})}_{\text{Value}} \underbrace{-\lambda_o \log p(\mathbf{x}|\mathbf{z})}_{\text{Originality}}. \quad (10)$$

5 Implementation and Optimization with Autoregressive Models

We now discuss the implementation of the CoVO score with autoregressive models. Using the notation introduced above, in the context of a $\boldsymbol{\theta}$-parameterized LLM, $p(\mathbf{x}|\mathbf{z})$ can be expressed as $\prod_{t=1}^{T} p_{\boldsymbol{\theta}}(x_t|\mathbf{x}_{<t},\mathbf{z})$. However, considering just the product of all the conditioned probabilities for an optimization problem would lead to preferring shorter sequences. To avoid this, we propose to use the T-th root: $\sqrt[T]{\prod_{t=1}^{T} p_{\boldsymbol{\theta}}(x_t|\mathbf{x}_{<t},\mathbf{z})}$. By leveraging the properties of the logarithm, we obtain:

$$\lambda_v \frac{\sum_{i=1}^{|\mathbf{z}|} \log p_{\boldsymbol{\theta}}(z_i|\mathbf{z}_{<i},\mathbf{x})}{|\mathbf{z}|} - \lambda_o \frac{\sum_{j=1}^{|\mathbf{x}|} \log p_{\boldsymbol{\theta}}(x_j|\mathbf{x}_{<j},\mathbf{z})}{|\mathbf{x}|}. \quad (11)$$

It is worth noting that the vocabulary of an LLM can be extremely large, which can cause $p_{\boldsymbol{\theta}}(a|b)$ to be small even when a is the most probable event given b. In particular, when an LLM generates $\mathbf{x}$ given $\mathbf{z}$ and then evaluates both $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$ and $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})$, this can lead to a significant discrepancy between the magnitude of value and originality.

Since $\mathbf{x}$ has been sampled from a model based on $p_{\boldsymbol{\theta}}$, its probability would be high by definition. However, there may be various ways (possibly through synonyms) to define $\mathbf{x}$, leading to a smaller probability of $\mathbf{z}$.

Inspired by Macedo et al. (2004), we propose to counteract this problem by normalizing $p_{\boldsymbol{\theta}}(a|b)$ via $n' = \frac{n-n_{min}}{n_{max}-n_{min}}$. For probabilities, $n_{min} = 0$, while $n_{max} = \max_{v \in \mathcal{V}} p_{\boldsymbol{\theta}}(b)$, thus obtaining the overall mapping for $p_{\boldsymbol{\theta}}$: $\frac{p_{\boldsymbol{\theta}}(x_t|\mathbf{x}_{<t},\mathbf{z})}{\max_{v \in \mathcal{V}} p_{\boldsymbol{\theta}}(\mathbf{x}_{<t},\mathbf{z})}$. Once again, by applying the properties of logarithms, we obtain:

$$s_{CoVO}^{AR_{norm}}(\mathbf{z}, \mathbf{x}, p_{\boldsymbol{\theta}}) = \lambda_v s_v(\mathbf{z}, \mathbf{x}, p_{\boldsymbol{\theta}}) + \lambda_o s_o(\mathbf{z}, \mathbf{x}, p_{\boldsymbol{\theta}}) = \lambda_v \frac{\sum_{i=1}^{|\mathbf{z}|}(\log p_{\boldsymbol{\theta}}(z_i|\mathbf{z}_{<i}, \mathbf{x}) - \max_{v \in \mathcal{V}} \log p_{\boldsymbol{\theta}}(\mathbf{z}_{<i}, \mathbf{x}))}{|\mathbf{z}|} - \lambda_o \frac{\sum_{j=1}^{|\mathbf{x}|}(\log p_{\boldsymbol{\theta}}(x_j|\mathbf{x}_{<j}, \mathbf{z}) - \max_{v \in \mathcal{V}} \log p_{\boldsymbol{\theta}}(\mathbf{x}_{<j}, \mathbf{z}))}{|\mathbf{x}|}. \quad (12)$$

Calculating $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})$ is not trivial. Since LLMs are trained to complete text sequences, it is unlikely that they would generate the source text immediately after the target text (which, we should remember, is generated immediately after the source text). To address this, we consider an approximation $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}')$, where $\mathbf{x}' = \mathbf{x} + \mathbf{q}$. Here, $\mathbf{q}$ represents an additional question, such as "How would you describe this text?" or a similar formulation designed solely to increase the likelihood of generating the source text $\mathbf{z}$ (as well as alternative sources).

Once the CoVO score has been defined, its adoption in an RL framework is straightforward. As previously introduced, we can directly utilize our CoVO score as the final reward for the generated sequence. Then, the model can be trained with any policy gradient method. Our experiments leverage GRPO (Shao et al., 2024), which is a state-of-the-art choice for training language models. As introduced above, GRPO adds a per-token KL divergence term to the loss rather than to the single rewards. Usually, the KL divergence is approximated with the following estimator: $r_{ref}(\boldsymbol{\theta}) - \log r_{ref}(\boldsymbol{\theta}) - 1$, where $r_{ref}(\boldsymbol{\theta}) = \frac{\pi_{ref}}{\pi_{\boldsymbol{\theta}}}$ and π_{ref} is the reference policy, i.e., the model before GRPO training. In particular, GRPO aims to *minimize* the KL divergence, thus the second term $-\log r_{ref} = -\log \pi_{ref} + \log \pi_{\boldsymbol{\theta}}$ can be seen as made of two additional optimization problems: the minimization of the originality component $-\log \pi_{ref}$ from Equation 10, and the maximization of surprisal, or self-information, $-\log \pi_{\boldsymbol{\theta}}$ under the current model. In other words, this second term somehow trades off a portion of the originality component under the reference model (proportional to the $\beta \ll 1$ coefficient) with an equally-weighted surprisal under the current model. However, the first term of the KL approximation keeps the two policies closer, preventing the trained one from deviating too much and potentially diminishing the impact of the originality component in favor of the value component.

6 Experiments

We evaluate the effectiveness of our RL strategy through three case studies: poetry generation, mathematical problem resolution, and the tasks included in NoveltyBench[1]. In all experiments, we employ two settings, i.e., GRPO to maximize the score from Equation 12 *without* the KL divergence loss, i.e., with $\beta = 0.0$ (CoVO); and GRPO to maximize the score from Equation 12 *with* the KL divergence loss, i.e., with $\beta = 0.05$ (CoVO+KL). Both methods assume $\lambda_v = \lambda_o = 1.0$. Finally, we restrict our baselines to the original model and, if available, a model tuned solely on the environmental reward. While other methods have been proposed recently to enhance the diversity of generated outputs (as introduced in Section 2.2), they still require an extrinsic reward, thus being incompatible with the majority of our experiments where no environmental reward is available. In addition, while it is common to induce diversity at the sampling level (Peeperkorn et al., 2024), our approach is orthogonal to the chosen sampling strategy and can benefit from more advanced methods. In our evaluation, we aim to verify whether our reward scheme can increase value and originality, regardless of how the output is sampled.

6.1 Poetry Generation

Experimental Setup. The first set of experiments concerns a very common creative task, aiming to teach the LLM to generate poems that are both more original and valuable. More specifically, we follow the approach outlined by Bradley et al. (2024) and instruct the model to write a poem in a particular style and tone. We adopt the `Meta-Llama-3-8B` (Llama Team, 2024) and `Qwen2.5-3B` (Qwen Team, 2025) models as our pre-trained agents. Since we do not use the

[1]The code and results of the experiments can be found at: https://anonymous.4open.science/r/CoVO-grpo/

Method	In-distribution				Out-of-distribution			
	Corr. ↑	Metric (L/S) ↑	T-LCS ↓	Tone ↑	Corr. ↑	Metric (L/S) ↑	T-LCS ↓	Tone ↑
`Meta-Llama-3-8B`	0.947	0.300 / 0.177	6.067 / 19	$0.664_{\pm 0.069}$	0.987	0.467 / 0.132	5.853 / 68	$0.622_{\pm 0.063}$
+CoVO	0.960	0.233 / 0.084	4.880 / 7	$0.749_{\pm 0.055}$	0.933	0.267 / 0.040	4.920 / 6	$0.665_{\pm 0.059}$
+CoVO+KL	0.920	0.367 / 0.068	5.093 / 9	$0.740_{\pm 0.058}$	0.960	0.311 / 0.117	5.160 / 22	$0.681_{\pm 0.059}$
`Qwen-2.5-3B`	0.973	0.500 / 0.105	5.080 / 8	$0.831_{\pm 0.053}$	0.960	0.600 / 0.113	4.813 / 8	$0.764_{\pm 0.062}$
+CoVO	0.920	0.500 / 0.156	4.893 / 7	$0.748_{\pm 0.064}$	0.947	0.578 / 0.102	4.800 / 7	$0.729_{\pm 0.062}$
+CoVO+KL	0.960	0.467 / 0.123	4.947 / 7	$0.787_{\pm 0.059}$	0.947	0.689 / 0.095	4.720 / 7	$0.721_{\pm 0.063}$

Table 1: Aggregate results of generated poems considering both training prompts (left) and testing prompts (right). Scores on the poetical metrics are reported at the line level (L) and syllable level (S) and only consider requests for styles with specific metrical properties. Under T-LCS, we report both the mean and the maximum longest common substring across all generated poems. The mean and the 95% confidence interval are reported for tone adherence.

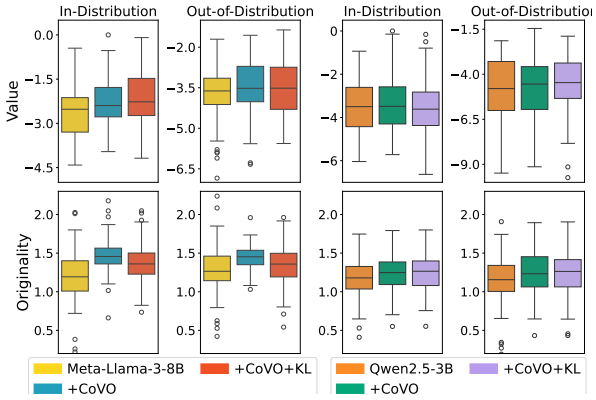

Figure 2: The distribution of value and originality scores for the in-distribution and out-of-distribution poems generated by the original models and our tuned versions.

instruction-tuned models, we prompt them with a few-shot examples of the task to make them more likely to produce the desired output in the desired form (the full prompt is reported in Appendix A, together with the training parameters). Instead of fine-tuning the entire network, we consider Low-Rank Adaptation (LoRA) (Hu et al., 2022). The original model is also used to compute the score. All sampling happens with top-k ($k = 50$) and at a temperature of 1.0. We perform a quantitative evaluation where we compute poetical metrics for quality (lexical correctness of poems, adherence to line- and syllable-level constraints, and tone adherence to the request through zero-shot classification with `bart-large-mnli` model (Lewis et al., 2020)) and for originality (accidental reproduction of existing poems). For the latter, we define a Token-based Longest Common Substring (T-LCS) score and use it to compare generated poems with a reference dataset of approx. 84k public-domain poems extracted from Project Gutenberg (please refer to Appendix B for a first presentation of our GutenVerse dataset). While a generated poem can be an accidental reproduction of a protected work or a different source (e.g., a song), we believe it can provide a useful evaluation tool to understand the general degree of originality.

Experimental Results. Table 1 reports the scores about the compliance of poetical constraints at the syllable and line levels, lexical correctness (the ratio of poems not containing noisy text), tone adherence (the zero-shot classification of that poem being of that tone rather than its opposite), and accidental reproduction rate (the mean and maximum token-based longest common substring).

Overall, our CoVO-based fine-tuning leads to a higher tone adherence for Llama-based models and a lower reproduction rate for both models, at the potential cost of metric adherence, especially without the KL loss. Indeed, its role seems to foster quality (especially in terms of metrical correctness), trading off some originality. On the contrary, not using the KL loss arguably avoids any significant reproduction, as demonstrated by the very low maximum token-based longest common substring.

Interestingly, these considerations align well with our CoVO score. Figure 2 reports the value and originality according to Equation 12 under the pre-trained model. While the two methods do not significantly differ from the baseline (which is possibly due to the opposite forces of value and originality (Varshney, 2019)), we again see that the presence of KL leads to slightly higher value, while its absence leads to slightly higher originality. However, aggregated scores, such as those presented here, might be insufficient. For a more complete overview, we also conducted a fine-grained analysis of the generated poems in Appendix C.1.

6.2 Math Problem Resolution

Experimental Setup. The second set of experiments concerns a more practical and quantitative task, as it aims to teach the LLM to solve

MetaMath-Mistral-7B	GSM8K					MATH				
	Accuracy	Cross-Input Diversity		Against-Pretrained Diversity		Accuracy	Cross-Input Diversity		Against-Pretrained Diversity	
		EAD	SBERT	EAD	SBERT		EAD	SBERT	EAD	SBERT
Original model	77.96%(3)	2.0071	0.6402	-	-	33.55%(483)	5.7239	**0.8032**	-	-
+Ext	78.18%(4)	<u>2.0045</u>	**0.6404**	$0.0081_{\pm.0021}$	$0.0008_{\pm.0002}$	<u>33.19</u>%(469)	<u>5.7187</u>	<u>0.8027</u>	$0.0333_{\pm.0029}$	$0.0074_{\pm.0008}$
+Ext+KL	78.08%(5)	2.0077	<u>0.6401</u>	$0.0096_{\pm.0024}$	$0.0011_{\pm.0004}$	33.56%(476)	5.7517	0.8028	$0.0345_{\pm.0030}$	$0.0075_{\pm.0008}$
+CoVO	78.12%(3)	**2.0509**	0.6403	$\mathbf{0.0854_{\pm.0062}}$	$\mathbf{0.0118_{\pm.0012}}$	33.29%(521)	5.8219	0.8029	$\mathbf{0.1457_{\pm.0049}}$	$\mathbf{0.0339_{\pm.0016}}$
+CoVO+KL	78.12%(3)	2.0464	0.6402	$\mathbf{0.0879_{\pm.0063}}$	$\mathbf{0.0122_{\pm.0013}}$	32.79%(533)	**5.8442**	0.8030	$\mathbf{0.1479_{\pm.0050}}$	$\mathbf{0.0343_{\pm.0016}}$
+CoVO+Ext	**78.33**%(4)	2.0340	**0.6404**	$0.0628_{\pm.0056}$	$0.0088_{\pm.0011}$	**33.76**%(503)	5.8114	0.8030	$0.1136_{\pm.0047}$	$0.0259_{\pm.0015}$
+CoVO+Ext+KL	<u>77.95</u>%(4)	2.0367	0.6402	$0.0638_{\pm.0057}$	$0.0089_{\pm.0011}$	33.74%(497)	5.8082	0.8031	$0.1102_{\pm.0047}$	$0.0250_{\pm.0015}$

Table 2: Accuracy and diversity of results for the GSM8K (left) and MATH (right) test sets. In brackets, the number of responses exceeding the maximum token limit. The best scores are highlighted in **bold**, while the worst scores are indicated with <u>underlining</u>. The mean and the 95% confidence interval are reported for against-pretrained diversity.

mathematical problems through more diverse procedures. In particular, we focus on the Mistral-based `MetaMath-Mistral-7B` model, i.e., fine-tuned with self-supervised learning on the MetaMathQA (Yu et al., 2024). It is a dataset of textual math questions paired with responses where the numerical answer is easily separable from the textual procedure. While the entire set contains 395k entries, making an additional training epoch too expensive, MetaMathQA is composed of entries from two different training sets, then augmented with various techniques: GSM8K (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021). Since we are only interested in the questions, we limit our training to a single epoch over those datasets. Moreover, we exclude all questions with a tokenized length of either question or answer greater than 512, obtaining 14876 out of 14973 total entries. We separate the procedure and the answer from each solution to train our model and use the numerical answer to check the correctness of the predicted solution. The RL problem can then be formulated considering up to two rewards: our CoVO score computed on the procedure and a verifiable, extrinsic reward based on the correctness of the answer. Instead of fine-tuning the entire model, we adopt a more parameter-efficient strategy with LoRA and use the original model to perform the CoVO score computation. Following Yu et al. (2024), the outputs are obtained with a greedy strategy.

The evaluation considers both GSM8K and MATH test sets (limited to the entries with a tokenized length of question and answer smaller than 512, i.e., all 1319 entries for GSM8K and 4546 out of 5000 for MATH). We compute the percentage of correct solutions together with two diversity metrics: expectation-adjusted distinct N-grams (EAD) (Liu et al., 2022) and sentence embedding cosine similarity (SBERT) (Reimers and Gurevych, 2019), which should measure syntactical and semantical diversity, respectively (Kirk et al., 2024). EAD counts the number of distinct N-grams (averaging over $N = 1 \ldots 5$) across all generated responses and removes the bias toward shorter inputs by scaling the number of distinct tokens based on their expectations. The SBERT metric computes the average of the cosine similarity between the embeddings of any possible pairs of outputs and returns 1 minus the similarity. While this was originally based on Sentence-BERT, we employ the more recent `all-mpnet-base-v2` instead, as suggested by their developers[2]. Following Kirk et al. (2024), we compute *cross-input* EAD and SBERT, i.e., we derive them by considering all outputs produced for a specific seed together. In addition, we also calculate *against-pretrained* EAD and SBERT. Given each input, we compare the output with the one from the pre-trained model by calculating the average expectation-adjusted distinct N-grams not present in the pre-trained model response, and 1 minus the cosine similarity between the two outputs, respectively.

Experimental Results. Table 2 reports the results for the GSM8K and MATH test sets. For the GSM8K test set, while all methods achieve similar results, using the CoVO score only (with and without the KL loss) leads to greater EAD diversity and to diverge more from the original model, while the presence of the math reward leads to greater accuracy, especially without KL.

The results for the MATH test confirm that the most accurate method is the one trained to optimize the CoVO score and the extrinsic reward, with a negligible trade-off in terms of diversity, since the cross-input EAD and the against-pretrained scores are still substantially higher than those from the baselines. However, removing the extrinsic reward likely pushes the model too far from its pre-trained

[2] https://huggingface.co/sentence-transformers/bert-large-nli-stsb-mean-tokens

version, causing the accuracy to decrease.

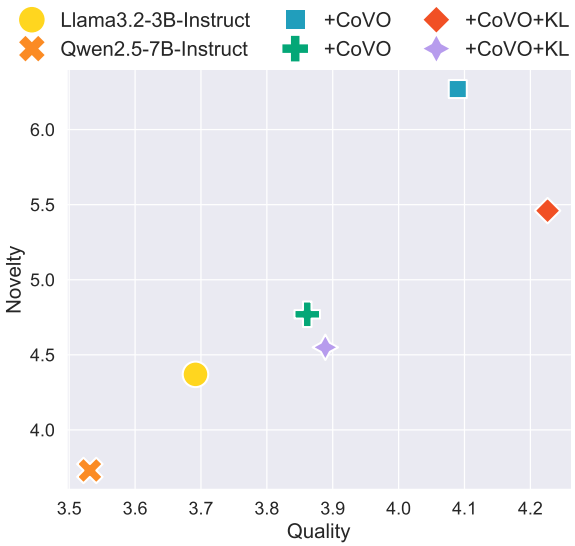

Figure 3: The mean utility (quality) and mean distinct (novelty) scores on NoveltyBench ('curated' partition) for the original models and our methods (tuned on the 'wildchat' partition).

6.3 NoveltyBench

Experimental Setup. Finally, we also experiment with NoveltyBench (Zhang et al., 2025), a recent benchmark that aims to evaluate the ability of language models to produce multiple distinct and high-quality outputs. NoveltyBench contains two sets of prompts thought for eliciting diverse responses: the 'curated' partition, with 100 prompts manually curated by the paper's authors, and the 'wildchat' partition, with 1000 prompts sourced from the WildChat-1M dataset (Zhao et al., 2024). The benchmark requires a language model to generate 10 outputs per prompt using a temperature setting of 1.0, after which it computes scores for novelty and quality. To assess novelty, the 10 outputs are grouped into equivalence classes using a fine-tuned DeBERTa model (He et al., 2021), and the number of distinct classes is reported as the novelty score. To assess quality, it computes the cumulative utility of the 10 outputs, where the utility is 0 if the i-th output has the same equivalence class as a precedent output, and a calibrated reward from `Skywork-Reward-Gemma-2-27B-v0.2` (Liu et al., 2024) otherwise (we refer to Zhang et al. (2025) for full details). We fine-tune the `Qwen2.5-7B-Instruct` (Qwen Team, 2025) and the `Llama-3.2-3B-Instruct` (Llama Team, 2024) models for a single epoch on the 'wildchat' partition (the full parameters are reported in Appendix A); then, we compute the novelty and quality scores on the 'curated' partition, and we compare them with the performance of the original model.

Experimental Results. Figure 3 reports the quality and novelty scores achieved by our methods, compared to those from the original instructed models. Optimizing for the CoVO score results in substantial improvements in both novelty and quality metrics for both `Llama-3.2-3B-Instruct` and `Qwen-2.5-7B` models, with the greatest gains in novelty. Moreover, these results underscore the interplay between KL loss and our CoVO score: incorporating the KL penalty tends to improve quality, but at the cost of (slightly) reduced novelty.

7 Conclusion

In this paper, we presented CoVO, a novel score that quantifies the value and originality of neurally-generated text. The definition of CoVO is based on the analysis of mutual information between the model's outputs and inputs, and vice versa. We also proposed an optimization problem where a generative model aims to maximize this score to generate more creative products, and detailed how to use it in language modeling. We conducted experiments on poetry generation, math problem solving, and tasks from NoveltyBench, exploring trade-offs in accuracy vs diversity. Effectively balancing value and originality maximization remains an open question, but our score seems to capture properties functionally aligned with creativity-relevant aspects of language generation, as demonstrated through both conceptual analysis and empirical results. When used as a reward function in GRPO, it promotes improvements in both the novelty and quality of model outputs, even with relatively few optimization steps, and without relying on external rewards, which are required by most alternative methods.

Our research agenda aims to extend our method to other models and tasks, to include inference-level strategies such as creativity-oriented sampling schemes, and to explore its use for evaluation (e.g., in a Best-of-N setting (Stiennon et al., 2020)) rather than solely for optimization. We also plan to investigate the definition of additional scores for capturing other potentially relevant aspects of the creative process. Despite being costly and inherently constrained (Davis, 2024), assessing whether our creativity score aligns with human judgment is another key direction for future work.

Limitations

There are a few limitations worth noting. Firstly, our score represents only a quantifiable approximation of a particular theoretical perspective on creativity, grounded in the dimensions of value and originality. For example, value has been considered solely from the perspective of effectiveness, while other dimensions have been proposed as well (e.g., interestingness (Boden, 1994) or monetary worth (Lepak et al., 2007)). Quality and correctness are two additional key dimensions of creative value (Boden, 2003). Although they can be indirectly incorporated through extrinsic environmental rewards (as in math problem-solving tasks) and the KL-penalty loss that preserves pre-training behaviors, they are not explicitly captured by our score. In addition, other classic definitions of creativity, such as that presented by Boden (2003), add a third requirement by splitting originality into novelty and surprise. Moreover, our score reflects a specific view of the evaluation of creativity based on the generated outputs and does not account for potential alternative theories (for example, arising from different cultures (Lubart, 1999)) and perspectives (Rhodes, 1961). From a practical point of view, computing the CoVO score requires an additional pass inside the model, which represents a potential bottleneck for larger LLMs. In addition, maximizing it can lead to reward hacking (Pan et al., 2022), where one of the two score components is exploited, e.g., only maximizing originality (outputting something completely unpredictable and meaningless) or only maximizing value (e.g., repeating the request, which is not original at all and helps hack the value-based reward function). We experienced a few of these failure modes when the CoVO score was the only optimization target (please refer to the detailed analysis in Appendix C.1). However, retaining both scores and incorporating the KL divergence largely mitigated these failure modes in our experiments. Finally, our evaluation is currently limited to only three relatively short-form text generation tasks. While their generalizability is supported by the theoretical framework discussed above, the resulting performance was experimentally evaluated for a finite number of scenarios.

Ethical Considerations

The authors are aware of the potential impact that generative technologies might have on the production of artistic outputs and, as a consequence, on human artists. This work may contribute to enhancing the quality of generated outputs. However, the authors argue that typical traits of human creativity, such as the active participation of artists in the creative process, cannot be directly replicated by machines, as pointed out by Runco in the updated standard definition of creativity (Runco, 2025). The authors refer interested readers to a previous work of theirs (*citation removed for double-blind submission*), in which these themes are discussed in detail.

References

Joshua Achiam and Shankar Sastry. 2016. Surprise-Based Intrinsic Motivation for Deep Reinforcement Learning. In *Proc. of the NeurIPS'16 Deep RL Workshop*.

Oron Anschel, Alon Shoshan, Adam Botach, Shunit Haviv Hakimi, Asaf Gendler, Emanuel Ben Baruch, Nadav Bhonker, Igor Kviatkovsky, Manoj Aggarwal, and Gerard Medioni. 2025. Group-aware reinforcement learning for output diversity in large language models. In *Proc. of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP'25)*.

Pierre Baldi and Laurent Itti. 2010. Of Bits and Wows: A Bayesian Theory of Surprise with Applications to Attention. *Neural Networks*, 23:649–666.

Andrew Barto, Marco Mirolli, and Gianluca Baldassarre. 2013. Novelty or surprise? *Frontiers in Psychology*, 4:907:1–907:15.

Margaret A. Boden. 1994. *Dimensions of Creativity*. The MIT Press.

Margaret A. Boden. 2003. *The Creative Mind: Myths and Mechanisms*. Routledge.

Herbie Bradley, Andrew Dai, Hannah Teufel, Jenny Zhang, Koen Oostermeijer, Marco Bellagente, Jeff Clune, Kenneth Stanley, Grégory Schott, and Joel Lehman. 2024. Quality-Diversity through AI feedback. In *Proc. of the 12th International Conference on Learning Representations (ICLR'24)*.

Razvan C. Bunescu and Oseremen O. Uduehi. 2019. Learning to surprise: A composer-audience architecture. In *Proc. of the 10th International Conference on Computational Creativity (ICCC'19)*.

Xiwen Chen, Wenhui Zhu, Peijie Qiu, Xuanzhao Dong, Hao Wang, Haiyu Wu, Huayu Li, Aristeidis Sotiras, Yalin Wang, and Abolfazl Razi. 2025. DRA-GRPO: Your GRPO needs to know diverse reasoning paths for mathematical reasoning. ArXiv:2505.09655 [cs.CL].

Daixuan Cheng, Shaohan Huang, Xuekai Zhu, Bo Dai, Wayne Xin Zhao, Zhenliang Zhang, and Furu Wei. 2026. Reasoning with exploration: An entropy perspective. In *Proc. of the 40th AAAI Conference on Artificial Intelligence (AAAI'26)*.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems (NeurIPS'17)*.

John Joon Young Chung, Vishakh Padmakumar, Melissa Roemmele, Yuqian Sun, and Max Kreminski. 2025. Modifying large language model post-training for diverse creative writing. In *Proc. of the 2nd Conference on Language Modeling (COLM'25)*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. ArXiv:2110.14168 [cs.LG].

Thomas M Cover. 1999. *Elements of Information Theory*. John Wiley & Sons.

Mihaly Csikszentmihalyi. 2014. Society, Culture, and Person: A Systems View of Creativity. In *The Systems Model of Creativity*, pages 47–71. Springer.

Runpeng Dai, Linfeng Song, Haolin Liu, Zhenwen Liang, Dian Yu, Haitao Mi, Zhaopeng Tu, Rui Liu, Tong Zheng, Hongtu Zhu, and Dong Yu. 2026. CDE: Curiosity-driven exploration for efficient reinforcement learning in large language models. In *Proc. of the 14th International Conference on Learning Representations (ICLR'26)*.

Ernest Davis. 2024. ChatGPT's poetry is incompetent and banal: A discussion of (Porter and Machery, 2024). https://cs.nyu.edu/~davise/papers/GPT-Poetry.pdf [Accessed July 31, 2025].

Anil R. Doshi and Oliver P. Hauser. 2024. Generative AI enhances individual creativity but reduces the collective diversity of novel content. *Science Advances*, 10(28):eadn5290.

Giorgio Franceschelli and Mirco Musolesi. 2024a. Creativity and machine learning: A survey. *ACM Computing Surveys*, 56(11):283:1–41.

Giorgio Franceschelli and Mirco Musolesi. 2024b. Reinforcement learning for generative AI: State of the art, opportunities and open research challenges. *Journal of Artificial Intelligence Research*, 79:417–446.

Giorgio Franceschelli and Mirco Musolesi. 2025. On the creativity of large language models. *AI & Society*, 40:3785–3795.

Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III, and Kate Crawford. 2018. Datasheets for datasets. *Communications of the ACM*, 64(12):86–92.

Andre Wang He, Daniel Fried, and Sean Welleck. 2025. Rewarding the unlikely: Lifting GRPO beyond distribution sharpening. In *Proc. of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP'25)*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. DeBERTa: Decoding-enhanced BERT with disentangled attention. In *Proc. of the 9th International Conference on Learning Representations (ICLR'21)*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the MATH dataset. In *Proc. of the 35th Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *Proc. of the 10th International Conference on Learning Representations (ICLR'22)*.

Mete Ismayilzada, Antonio Laverghetta Jr., Simone A. Luchini, Reet Patel, Antoine Bosselut, Lonneke Van Der Plas, and Roger E. Beaty. 2025. Creative preference optimization. In *Findings of the Association for Computational Linguistics (EMNLP'25)*.

Robert Kirk, Ishita Mediratta, Christoforos Nalmpantis, Jelena Luketina, Eric Hambro, Edward Grefenstette, and Roberta Raileanu. 2024. Understanding the effects of RLHF on LLM generalisation and diversity. In *Proc. of the 12th International Conference on Learning Representations (ICLR'24)*.

Arash Lagzian, Srinivas Anumasa, and Dianbo Liu. 2025. Multi-novelty: Improve the diversity and novelty of contents generated by large language models via inference-time multi-views brainstorming. ArXiv:2502.12700 [cs.CL].

Carolyn Lamb, Daniel G. Brown, and Charles L. A. Clarke. 2019. Evaluating Computational Creativity: An Interdisciplinary Tutorial. *ACM Computing Surveys*, 51(2):28:1–34.

Byung Cheol Lee and Jaeyeon (Jae) Chung. 2024. An empirical investigation of the impact of ChatGPT on creativity. *Nature Human Behaviour*, 8:1906–1914.

David P. Lepak, Ken G. Smith, and M. Susan Taylor. 2007. Value creation and value capture: A multilevel perspective. *Academy of Management Review*, 32(1):180–194.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training

for natural language generation, translation, and comprehension. In *Proc. of the 58th Annual Meeting of the Association for Computational Linguistics (ACL'20)*.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *Proc. of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL'16)*.

Tianjian Li, Yiming Zhang, Ping Yu, Swarnadeep Saha, Daniel Khashabi, Jason Weston, Jack Lanchantin, and Tianlu Wang. 2025. Jointly reinforcing diversity and quality in language model generations. ArXiv:2509.02534 [cs.CL].

Zhiyu Lin, Upol Ehsan, Rohan Agarwal, Samihan Dani, Vidushi Vashishth, and Mark Riedl. 2023. Beyond prompts: Exploring the design space of mixed-initiative co-creativity systems. In *Proc. of the 14th International Conference on Computational Creativity (ICCC'23)*.

Chris Yuhao Liu, Liang Zeng, Jiacai Liu, Rui Yan, Jujie He, Chaojie Wang, Shuicheng Yan, Yang Liu, and Yahui Zhou. 2024. Skywork-Reward: Bag of tricks for reward modeling in LLMs. ArXiv:2410.18451 [cs.AI].

Siyang Liu, Sahand Sabour, Yinhe Zheng, Pei Ke, Xiaoyan Zhu, and Minlie Huang. 2022. Rethinking and refining the distinct metric. In *Proc. of the 60th Annual Meeting of the Association for Computational Linguistics (ACL'22)*.

AI @ Meta Llama Team. 2024. The Llama 3 herd of models. ArXiv:2407.21783 [cs.AI].

Li-Chun Lu, Miri Liu, Pin Chun Lu, Yufei Tian, Shao-Hua Sun, and Nanyun Peng. 2026. Rethinking creativity evaluation: A critical analysis of existing creativity evaluations. In *Proc. of the 19th Conference of the European Chapter of the Association for Computational Linguistics (EACL'26)*.

Ximing Lu, Melanie Sclar, Skyler Hallinan, Niloofar Mireshghallah, Jiacheng Liu, Seungju Han, Allyson Ettinger, Liwei Jiang, Khyathi Chandu, Nouha Dziri, and Yejin Choi. 2025. AI as humanity's Salieri: Quantifying linguistic creativity of language models via systematic attribution of machine text against web text. In *Proc. of the 13th International Conference on Learning Representations (ICLR'25)*.

Todd I. Lubart. 1999. Creativity across cultures. In *Handbook of Creativity*, pages 339–350. Cambridge University Press.

Luís Macedo, Rainer Reisenzein, and Amílcar Cardoso. 2004. Modeling forms of surprise in artificial agents: empirical and theoretical study of surprise functions. In *Proceedings of the Annual Meeting of the Cognitive Science Society (CogSci'04)*.

David J.C. MacKay. 2003. *Information Theory, Inference and Learning Algorithms*. Cambridge University Press.

Nguyen Nhat Minh, Andrew Baker, Clement Neo, Allen G Roush, Andreas Kirsch, and Ravid Shwartz-Ziv. 2025. Turning up the heat: Min-p sampling for creative and coherent LLM outputs. In *Proc. of the Thirteenth International Conference on Learning Representations (ICLR'25)*.

Alexander Pan, Kush Bhatia, and Jacob Steinhardt. 2022. The effects of reward misspecification: Mapping and mitigating misaligned models. In *Proc. of the International Conference on Learning Representations (ICLR'22)*.

Rafael Pardinas, Gabriel Huang, David Vazquez, and Alexandre Piché. 2023. Leveraging human preferences to master poetry. In *Proc. of the AAAI'23 Workshop on Creative AI Across Modalities*.

Max Peeperkorn, Tom Kouwenhoven, Dan Brown, and Anna Jordanous. 2024. Is temperature the creativity parameter of large language models? In *Proc. of the 15th International Conference on Computational Creativity (ICCC'24)*.

Brian Porter and Edouard Machery. 2024. AI-generated poetry is indistinguishable from human-written poetry and is rated more favorably. *Scientific Reports*, 14(1):26133.

Erfan Baghaei Potraghloo, Seyedarmin Azizi, Souvik Kundu, and Massoud Pedram. 2025. Top-H decoding: Adapting the creativity and coherence with bounded entropy in text generation. In *Proc. of the 39th Annual Conference on Neural Information Processing Systems (NeurIPS'25)*.

Qwen Team. 2025. Qwen2.5 technical report. ArXiv:2412.15115 [cs.CL].

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proc. of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP'19)*.

Mel Rhodes. 1961. An Analysis of Creativity. *The Phi Delta Kappan*, 42(7):305–310.

Baptiste Rozière, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Romain Sauvestre, Tal Remez, Jérémy Rapin, Artyom Kozhevnikov, Ivan Evtimov, Joanna Bitton, Manish Bhatt, Cristian Canton Ferrer, Aaron Grattafiori, Wenhan Xiong, Alexandre Défossez, and 7 others. 2023. Code Llama: Open Foundation Models for Code. ArXiv:2308.12950 [cs.CL].

Mark A. Runco. 2025. Updating the standard definition of creativity to account for the artificial creativity of AI. *Creativity Research Journal*, 37(1):1–5.

Mark A. Runco and Garrett J. Jaeger. 2012. The standard definition of creativity. *Creativity Research Journal*, 24(1):92–96.

Arkadiy Saakyan, Najoung Kim, Smaranda Muresan, and Tuhin Chakrabarty. 2026. Death of the novel(ty): Beyond N-gram novelty as a metric for textual creativity. In *Proc. of the 14th International Conference on Learning Representations (ICLR'26)*.

Piotr Sawicki, Marek Grzés, Fabricio Goes, Dan Brown, Max Peeperkorn, Aisha Khatun, and Simona Paraskevopoulou. 2023. Bits of Grass: Does GPT already know how to write like Whitman? In *Proc. of the 14th International Conference on Computational Creativity (ICCC'23)*.

John Schulman, Philipp Moritz, Sergey Levine, Michael I. Jordan, and Pieter Abbeel. 2016. High-dimensional continuous control using generalized advantage estimation. In *Proc. of the 4th International Conference on Learning Representations (ICLR'16)*.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal Policy Optimization Algorithms. ArXiv:1707.06347 [cs.LG].

Chantal Shaib, Venkata S Govindarajan, Joe Barrow, Jiuding Sun, Alexa Siu, Byron C Wallace, and Ani Nenkova. 2025. Standardizing the measurement of text diversity: A tool and comparative analysis. In *Proc. of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (AACL'25)*.

Claude Elwood Shannon. 1948. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. ArXiv:2402.03300 [cs.CL].

Chenglei Si, Diyi Yang, and Tatsunori Hashimoto. 2025. Can LLMs generate novel research ideas? a large-scale human study with 100+ NLP researchers. In *Proc. of the 13th International Conference on Learning Representations (ICLR'25)*.

Stewart Slocum, Asher Parker-Sartori, and Dylan Hadfield-Menell. 2025. Diverse preference learning for capabilities and alignment. In *Proc. of the 13th International Conference on Learning Representations (ICLR'25)*.

Claire Stevenson, Iris Smal, Matthijs Baas, Raoul Grasman, and Han van der Maas. 2022. Putting GPT-3's creativity to the (Alternative Uses) Test. In *Proc. of the 13th International Conference on Computational Creativity (ICCC'22)*.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to summarize with human feedback. In *Advances in Neural Information Processing Systems (NeurIPS'20)*.

Haoran Sun, Yekun Chai, Shuohuan Wang, Yu Sun, Hua Wu, and Haifeng Wang. 2025. Curiosity-driven reinforcement learning from human feedback. In *Proc. of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL'25)*.

Myron Tribus. 1961. *Thermodynamics and Thermostatics: An Introduction to Energy, Information and States of Matter, with Engineering Applications*. Van Nostrand.

Sergey Troshin, Wafaa Mohammed, Yan Meng, Christof Monz, Antske Fokkens, and Vlad Niculae. 2025. Control the temperature: Selective sampling for diverse and high-quality LLM outputs. In *Proc. of the 2nd Conference on Language Modeling (COLM'25)*.

Praveen Subbiah Vadivel. 2026. The algorithmic muse: Can AI truly master creative writing? *Communications of the ACM*, 69(3):20–23.

Lav R. Varshney. 2019. Mathematical limit theorems for computational creativity. *IBM Journal of Research and Development*, 63(1):2:1–2:12.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng YU, Zhengying Liu, Yu Zhang, James Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. 2024. MetaMath: Bootstrap Your Own Mathematical Questions for Large Language Models. In *Proc. of the 12th International Conference on Learning Representations (ICLR'24)*.

Yiming Zhang, Harshita Diddee, Susan Holm, Hanchen Liu, Xinyue Liu, Vinay Samuel, Barry Wang, and Daphne Ippolito. 2025. NoveltyBench: Evaluating creativity and diversity in language models. In *Proc. of the 2nd Conference on Language Modeling (COLM'25)*.

Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. 2024. WildChat: 1M ChatGPT interaction logs in the wild. In *Proc. of the 12th International Conference on Learning Representations (ICLR'24)*.

A Implementation Details

Parameter	Value
Total batches	100
Batch size B	4
Gradient accumulation steps	8
Max new tokens	256
Temperature	1.
Top-k	0
Optimizer	Adam
Learning rate	1e-5
Max gradient normalization	100.
Rank (LoRA)	16
α parameter (LoRA)	32
Dropout (LoRA)	0.05
Training iterations	1
Scale rewards	True
β (when used)	0.05
Number of generations G	4

Table 3: Training parameters for poetry generation.

The experiments were carried out using a Linux-based local server with two 80GB NVIDIA H100 GPUs, running Python 3.11.9. All the trainings were conducted with a random seed equal to 1 (set through the `set_seed` method from the HuggingFace `transformers` library), while poems were sampled at inference time with three different seeds (1, 42, and 121). The hyperparameters were selected to accommodate the most efficient use of the available resources, and otherwise set according to their default values from HuggingFace `transformers` and `trl` libraries. Only the learning rate has been tuned for the different tasks according to their training performances.

Poetry Generation. Table 3 reports the full training parameters for the experiments on poetry generation. The prompt for generation leverages *Nothing gold can stay* by Robert Frost, *Fame is a bee* by Emily Dickinson, and *Epitaph* by William Carlos Williams for few-shot learning:

> Write a fatalistic epigram poem of high, award winning quality.
>
> Nature's first green is gold,
> Her hardest hue to hold.
> Her early leaf's a flower;
> But only so an hour.
> Then leaf subsides to leaf.
> So Eden sank to grief,
> So dawn goes down to day.
> Nothing gold can stay.
>
> Write an ironic quatrain poem of high, award winning quality.
>
> Fame is a bee.
> It has a song-
> It has a sting-
> Ah, too, it has a wing.
>
> Write a naturalistic epitaph poem of high, award winning quality.
>
> An old willow with hollow branches
> Slowly swayed his few high fright tendrils
> And sang:
>
> Love is a young green willow
> Shimmering at the bare wood's edge.
>
> Write a `{tone}` `{style}` of high, award winning quality.

The training phase includes requests with tone-style pairs sampled among 'dark', 'happy', 'mysterious', 'reflective' or 'romantic' for the tone, and 'ballad', 'haiku', 'hymn', 'limerick' or 'sonnet' for the style, as done by Bradley et al. (2024). At inference time, we also consider 'cinquain', 'couplet', 'free verse', 'ode' or 'tanka' as styles and 'cutting', 'nostalgic', 'poignant', 'solemn' or 'whimsical' as tones. Instead, the prompt used for computing $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})$ is:

> Describe the style of the following poem in two words:
>
> `{poem}`
>
> I would describe it as a

Finally, the zero-shot classification for the tone adherence is performed with the given tone and 'not' plus the given tone as the candidate labels (e.g., if the required tone is 'happy', the two labels are ['happy', 'not happy']).

Math Problem Solving. Table 4 reports the full training parameters for math problem resolution. We also adopted the same two different prompts from Yu et al. (2024), i.e.:

Parameter	Value
Total epochs	1
Batch size B	4
Gradient accumulation steps	8
Max new tokens	512
Temperature	1.
Top-k	0
Optimizer	Adam
Learning rate	1e-6
Max gradient normalization	100.
Rank (LoRA)	16
α parameter (LoRA)	32
Dropout (LoRA)	0.05
Training iterations	1
Scale rewards	True
β (when used)	0.05
Number of generations G	4
Reward for correct answer	+1.

Table 4: Training parameters for math problem solving.

Parameter	Value
Total epochs	1
Batch size B	4
Gradient accumulation steps	8
Max new tokens	512
Temperature	1.
Top-k	0
Optimizer	Adam
Learning rate	1e-4
Max gradient normalization	100.
Rank (LoRA)	16
α parameter (LoRA)	32
Dropout (LoRA)	0.05
Training iterations	1
Scale rewards	True
β (when used)	0.05
Number of generations G	4

Table 5: Training parameters for NoveltyBench.

```
Below is an instruction that describes a task. Write a
response that appropriately completes the request.

### Instruction:
{question}

### Response:
```

at training time, and:

```
Below is an instruction that describes a task. Write a
response that appropriately completes the request.

### Instruction:
{question}

### Response: Let's think step by step.
```

at inference time. Instead, for computing $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})$ we used the following:

```
Below is a response that appropriately completes a
request. Write the instruction that describes the task.

### Response:
{response}

### Instruction:
```

NoveltyBench. Table 5 reports the full training parameters for NoveltyBench. At training and inference time, we simply adopt the following prompt:

```
user

{prompt}
assistant
```

where `user` and `assistant` are keywords used by the model to identify different roles in the chat. Instead, for computing $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})$ we used the following:

```
Below is a response that appropriately solves a task.
Write the instruction that describes the task.

### Response:
{response}

### Instruction:
```

B GutenVerse Dataset

To evaluate the accidental reproduction rate of generated poems, we introduce the GutenVerse dataset, which comprises over 84,000 public-domain, English-written poems extracted from Project Gutenberg. While generated poems can reproduce different content, e.g., songs or copyrighted material, we believe this can provide a useful indication of how likely a text is to be original or not.

To derive our dataset, we started from *Gutenberg, dammit*[3], a corpus of every plaintext file in Project Gutenberg (up until June 2016). We selected all

[3] https://github.com/aparrish/gutenberg-dammit/

text files whose metadata report English as the language, public domain as the copyright status, *poetry* among the subjects, or *poems* or *poetical work* in the title, and that were not translations of another book. Then, we applied a series of rules (e.g., about the verse length) to automatically extract the titles and poems from all the selected text files, and we defined our GutenVerse dataset. While it can still contain content that is not poetry (e.g., a table of contents formatted very uncommonly), the poems can be effectively used to measure the overlap between real and generated text.

We also released a datasheet (Gebru et al., 2018) for the GutenVerse dataset that can be found, together with the code used to create it, at: `https://anonymous.4open.science/r/GutenVerse-DD32/`.

C Qualitative Examples

In the following sections, we discuss the generated outputs for all our case studies from a qualitative perspective. While we present only a few particularly significant examples here, all outputs, together with their relative scores, are available at: `https://anonymous.4open.science/r/CoVO-grpo`.

C.1 Detailed Analysis of the Generated Poems

We now present some noteworthy generated poems to provide a detailed qualitative discussion of our methods and our score.

The baseline model `Meta-Llama-3-8B`, while producing prosaic text (without line breaks) occasionally, is the most conservative method. This results in its poems being usually well-formatted but also somehow *banal*, and more prone to reproducing existing works. For example, when asked to write a dark hymn, it produced the very famous two lines "I am the master of my fate: / I am the captain of my soul". In terms of the CoVO score, it is noteworthy that the poem with the lowest originality component achieves the highest T-LCS score. This strongly suggests that the model has memorized the hymn *All Things Bright and Beautiful* by Cecil Frances Alexander and reproduced it almost entirely, with only minor alterations in punctuation (enough to prevent the LCS score from being even higher). We report it in Table 6. Instead, for the overall score, the value part becomes the most relevant: the adherence to the requirement seems to be the reason behind the highest and lowest scores, as shown in Table 7.

Regarding the fine-tuned models, the one trained without the KL loss exhibits the opposite behavior. It never reproduces existing poems, but occasionally exploits the CoVO score in adversarial ways—for example, by learning to generate specific words associated with the requested tone. In one instance, it even learned to simply repeat the style-tone prompt itself, which resulted in the highest possible score, albeit at the cost of very low originality. Instead, maximizing the originality component makes the model produce noteworthy outputs, but without satisfying stylistic constraints, as reported in Table 8. In general, optimizing the CoVO score without the KL loss leads the model to deviate from memorized poems but also from metrical rules, and leads it to focus on the tone (both the explicit, requested tone and that intrinsic to the poetic style). For example, its limericks maintain the typical playful tone, but only once adhere to the traditional "There [once] was a man" starting line and only partially, as it starts with "There was an old dog". Interestingly, it is still not the highest-valued limerick; the best in terms of value component better shares the jocking nature of limericks (see Table 9).

On the contrary, including the KL loss makes the model adhere more to the required style; for instance, the majority of limericks start with different subjects of the classic "There [once] was a man". In general, the KL loss seems to help the model preserve coherence and increase the CoVO score in a more meaningful way, e.g., the repetitions are quite rare (though, when present, they lead again to the highest value scores), as well as the regurgitation of existing poems. Interestingly, this method also leads to the highest-scoring poem from an out-of-distribution tone-style pair, as reported in Table 10, together with the best poem that does not contain adversarial exploitation of the scores (e.g., repetitions or mention of the request), which, though not perfectly adhering with the syllable counts, share all the features of a proper haiku.

The qualitative findings for the `Qwen2.5-3` model and its CoVO and CoVO+KL tuned versions are quite similar, even though they are influenced by the different capabilities of the base model. Indeed, it never produces prosaic text, and its poems are better-formed. However, it tends to repeat the same sentences multiple times, leading to outputs that look more like poems but carry less meaning. To exemplify this, Table 11 reports the poem with the highest value score and the poem with the high-

est overall score. The first one, while seemingly well-formed, contains several repetitions; on the contrary, the second one does not satisfy the metrical constraints of a sonnet, but contains a language that seems very appropriate for it.

Finally, the `Qwen2.5-3` models tuned with the CoVO score (alone or with KL) behave similarly to those reported before. In particular, they sometimes exploit the score in an adversarial way, repeating sentences multiple times. However, this is occasionally incorporated in the poem in a positive way. For example, both the model tuned with the CoVO score alone and the model tuned with CoVO+KL produced two limericks that, despite containing several repetitions of the same items, are still meaningful and correctly depict the general spirit of limericks (see Table 12).

C.2 Math Problem Solving Output Exemplars

We conducted a qualitative analysis of the solutions provided by our methods and the baselines for the two math datasets. As also summarized by the quantitative metrics reported in the main paper, we found that the use of CoVO score, especially without KL, moves the model away from its most probable solutions, while the adoption of the external reward alone (with or without KL) is often not sufficient to alter the most probable answer in a meaningful way. However, the resulting solutions are highly similar across all cases. This suggests that the effect of the CoVO score is to redistribute the model's probability mass away from the originally most probable solutions and toward closely related, high-probability alternatives, in which only specific parts of the chain of thought are modified. While this can also impact the final numerical answer (as reported in the example from Table 13), it often means that the output has switched between alternative paths toward the same final response (as in Table 14).

C.3 NoveltyBench Output Analysis

A careful analysis of the generated outputs and the relative scores from NoveltyBench reveals that the impact of the CoVO score is particularly evident at the form level. Indeed, the answers to the questions are often semantically similar, sharing the same objects, but the sentences are formed differently: wherever the baseline would follow one or two templates, our training pushes the model to adopt multiple templates. This happens for both models we tested. As far as the `Llama3.2-3B-Instruct` model is concerned, Table 15 reports two examples of this. However, sometimes the CoVO score helps the model output a greater variety of semantic options as well, covering more exemplars of the same element (as in the first example from Table 16, where our methods return a larger number of different sites) or diverging from the most common terms (as in the second example from Table 16, where our methods produce more titles without banal terms such as 'translation', 'bits', and their variants).

The same considerations also apply to `Qwen2.5-7B-Instruct`, where these effects are sometimes even more apparent due to its lower output diversity, which often leads to identical or only slightly varied responses across runs. Here, the effect of the CoVO score is to yield outputs that follow alternative templates, but they are also characterized by distinct items or themes. Tables 17 and 18 report two examples that clearly manifest these features: in the first case, all the baseline outputs are identical, while our methods produce outputs that share the same templates (potentially slightly modified) but with two or three different content; in the second case, all the baseline outputs are again the same, while our methods produce outputs that diverge both in terms of semantic content and in terms of used templates.

Happy hymn	
all things bright and beautiful, all creatures great and small, all things wise and wonderful; the lord god made them all. each little flower that opens, each little bird that sings, he made their glowing colors, and spread their little wings. the rich man in his castle, the poor man at his gate, god made them high or lowly, and ordered their estate. the purple headed mountain, the river running by, the sunset and the morning that brightens up the sky;	
V = -2.118	O = 0.219

Table 6: The poem generated by the `Meta-Llama-3-8B` model with the highest T-LCS score and the lowest originality score.

Romantic sonnet	
when to the bat and the moth the world becomes a moon of beauty as twilight's cloak unfurls, and dreams and dreams and more dreams fill all of this fair air with poetry, a kiss is worth more than diamonds pure or sapphires rich and glinting like eyes of lovers in romance, where love is just like dew on wings of swans; and all that is beautiful is sweet, so love me as our love is new, my dear, for i love you!	
V = -0.449	O = 1.568
Nostalgic ode	
there was an old tree growing in a wood, its fruit was red, and its leaves were blue. an ant ran up it, and then down again, it turned into a purple snail in june.	
V = -6.845	O = 1.206

Table 7: The poems generated by the `Meta-Llama-3-8B` model with the highest (top) and lowest (bottom) overall CoVO score (top).

Mysterious haiku	
when no one sees, a child plays. i love what could be. no one believes? so they know her mind: so the bird sings for me alone this evening of dark rain?	
V = -2.501	O = 2.177
Whimsical couplet	
we need no art for that; but in wild nature see so sweet, it is a gift sang at every bird and flower.	
V = -2.103	O = 1.960

Table 8: The poems generated by the CoVO-based fine-tuned `Meta-Llama-3-8B` model with the highest originality scores from training (top) and testing (bottom) tone-style pairs.

Happy limerick	
The young man was a master of the rhyme and in love with the young lass to this time. So for her first night with her he wrote a new form to try in order the note… A true Limerick for her delight. Oh so happy and joyous the first night of love!	
V = -0.818	O = 1.370
Dark limerick	
There was an old dog from Japan Who fell in love with a harem of cans. He would hide there on hot afternoons, Till the sun turned his hair purple fawns. One day, while watching a game in blue jeans, He dropped by mistake the key to his knees. He gave the other a quick whiplash and then, Said to himself, "Ouch! my balls are in danger!" [...]	
V = -1.261	O = 1.354

Table 9: Two limericks from the CoVO-based fine-tuned `Meta-Llama-3-8B` model: above, the one with the highest value score; below, the only generated one that starts with the classic first line.

Romantic haiku	
spring winds carry in the leaves. love whispers.	
V = -1.343	O = 2.049
Whimsical couplet	
no more of one or two let there be more. one's never new but two and more's a rarity.	
V = -1.361	O = 1.718

Table 10: Two of the highest-scoring poems from the CoVO+KL-based fine-tuned `Meta-Llama-3-8B` model.

Romantic hymn	
hymn, for she, in his arms, he is loved, loved in every way, the beauty and the joy the glory that his life was. the one who loves is gentle, for her love for him is pure, the love that moves in and out of her heart a beautiful melody. for all the gifts of love, the wonders wrought, for all this beauty of love, his love for her is unending, the love that he holds close. for all the gifts of beauty, the wonders wrought, for all this glories love of her. the one who loves is gentle, for her love for him is pure.[...]	
V = -0.935	O = 0.864
Romantic sonnet	
how can i fail to love thee when thou are thou how can i forget such magic when eyes are blue how can i resist the call of thy when heart is my truest desire how shall i ever live without thee when thou art mine.	
V = -1.147	O = 1.224

Table 11: The poems generated by the `Qwen2.5-3` model with the highest value score (top) and highest overall score (bottom).

Romantic limerick	
A love that's not A love that's not A love that's not for the birds Is not A love, my birds.	
V = -2.169	O = 0.819
Romantic limerick	
A love that blooms and blooms, and brings back home alone; In the woods she waits, With each branch it's meant, A love that blooms and blooms.	
V = -1.831	O = 1.390

Table 12: Two examples of how the tuned versions of the `Qwen2.5-3` model (top, tuned with CoVO alone; bottom, tuned with CoVO+KL) correctly and positively exploit repetitions to write meaningful poems.

Question	Joe has $50 to buy an outfit for his new field trip. There is a 30% off sale at the clothing store. The shirt he picks out has a price of $25. He also picks out a pair of shorts for $35. Assuming that sales tax is included, how much money will Joe have left after the purchase? Answer: 8	
`MetaMath-Mistral-7`	First, Joe has $50 to spend. The shirt is 30% off, so it is discounted by 30/100 * $25 = $7.50 The discounted price of the shirt is $25 - $7.50 = $17.50 The shorts are $35. The total cost of the outfit is $17.50 + $35 = $52.50 Joe will have $50 - $52.50 = -$2.50 left after the purchase. Since Joe will have a negative amount left, it means he will not have any money left after the purchase. #### 0	
	Against-greedy EAD: 0.000	Against-greedy SBERT: 0.000
+ Ext. reward	First, Joe has $50 to spend. The shirt is 30% off, so it is discounted by 30/100 * $25 = $7.50 The discounted price of the shirt is $25 - $7.50 = $17.50 The shorts are $35. The total cost of the outfit is $17.50 + $35 = $52.50 Joe will have $50 - $52.50 = -$2.50 left after the purchase. Since Joe will have a negative amount left, it means he will not have any money left after the purchase. #### 0	
	Against-greedy EAD: 0.000	Against-greedy SBERT: 0.000
+ Ext + KL	First, Joe has $50 to spend. The shirt is 30% off, so it is discounted by 30/100 * $25 = $7.50 The discounted price of the shirt is $25 - $7.50 = $17.50 The shorts are $35. The total cost of the outfit is $17.50 + $35 = $52.50 Joe will have $50 - $52.50 = -$2.50 left after the purchase. Since Joe will have a negative amount left, it means he will not have any money left after the purchase. #### 0	
	Against-greedy EAD: 0.000	Against-greedy SBERT: 0.000
+ CoVO	First, Joe has $50 to spend. The shirt he picks out is $25, so he will spend $25 on the shirt. The pair of shorts he picks out is $35, so he will spend $35 on the shorts. The total amount he will spend is $25 + $35 = $60. Since there is a 30% off sale, Joe will get 30% off of $60, which is $60 * 0.30 = $18 off. So, the total amount he will pay after the discount is $60 - $18 = $42. Therefore, Joe will have $50 - $42 = $8 left after the purchase. #### 8	
	Against-greedy EAD: 0.524	Against-greedy SBERT: 0.108
+ CoVO + KL	Joe has $50 to spend. There is a 30% off sale, so he will pay 100% - 30% = 70% of the original price. The shirt is $25, so he will pay 70% of $25 = $17.50 for the shirt. The shorts are $35, so he will pay 70% of $35 = $24.50 for the shorts. The total cost of the outfit is $17.50 + $24.50 = $42. Joe will have $50 - $42 = $8 left after the purchase. #### 8	
	Against-greedy EAD: 0.377	Against-greedy SBERT: 0.093
+ CoVO + Ext	First, Joe has $50 to spend. The shirt he picks out is $25, so he will spend $25 on the shirt. The pair of shorts he picks out is $35, so he will spend $35 on the shorts. The total amount he will spend is $25 + $35 = $60. Since there is a 30% off sale, Joe will get 30% off of $60, which is $60 * 0.30 = $18 off. So, the total amount he will pay after the discount is $60 - $18 = $42. Therefore, Joe will have $50 - $42 = $8 left after the purchase. #### 8	
	Against-greedy EAD: 0.524	Against-greedy SBERT: 0.108
+ CoVO + Ext + KL	First, Joe has $50 to spend. The shirt he picks out is $25, so that leaves him with $50 - $25 = $25. Next, he picks out a pair of shorts for $35, so that leaves him with $25 - $35 = -$10. Since Joe has -$10 left, it means he spent more than he had. Therefore, Joe will have $0 left after the purchase. #### 0	
	Against-greedy EAD: 0.443	Against-greedy SBERT: 0.145

Table 13: A qualitative example of GSM8K outputs for the base `MetaMath-Mistral-7` model and its tuned variants with different responses, both in terms of form and of content.

Question	Jane has saved $4 of her allowance every week for the past 8 weeks. If she continues to save the same amount each week, how many more weeks will it take for her to have saved a total of $60?	
	Answer: 7	
`MetaMath-Mistral-7`	Jane has saved $4 every week for 8 weeks, so she has saved a total of $4 x 8 = $32. She wants to save a total of $60, so she still needs to save $60 - $32 = $28. If she continues to save $4 every week, it will take her $28 / $4 = 7 more weeks to save $28. Therefore, it will take her 7 more weeks to have saved a total of $60. #### 7	
	Against-greedy EAD: 0.000	Against-greedy SBERT: 0.000
+ Ext. reward	Jane has saved $4 every week for 8 weeks, so she has saved a total of $4 x 8 = $32. She wants to save a total of $60, so she still needs to save $60 - $32 = $28. If she continues to save $4 every week, it will take her $28 / $4 = 7 more weeks to save $28. Therefore, it will take her 7 more weeks to have saved a total of $60. #### 7	
	Against-greedy EAD: 0.000	Against-greedy SBERT: 0.000
+ Ext + KL	Jane has saved $4 every week for 8 weeks, so she has saved a total of $4 x 8 = $32. She wants to save a total of $60, so she still needs to save $60 - $32 = $28. If she saves $4 every week, it will take her $28 / $4 = 7 more weeks to save $28. #### 7	
	Against-greedy EAD: 0.036	Against-greedy SBERT: 0.012
+ CoVO	Jane has saved $4 every week for the past 8 weeks, so she has saved a total of $4 x 8 = $32. If she continues to save the same amount each week, we can set up an equation to find out how many more weeks it will take for her to have saved a total of $60. The equation is: $32 + $4x = $60, where x represents the number of additional weeks. To solve for x, we subtract $32 from both sides of the equation: $4x = $60 - $32. Simplifying the right side of the equation, we have: $4x = $28. To isolate x, we divide both sides of the equation by $4: x = $28 / $4. Simplifying the right side of the equation, we have: x = 7. Therefore, it will take Jane 7 more weeks to have saved a total of $60. #### 7	
	Against-greedy EAD: 0.433	Against-greedy SBERT: 0.049
+ CoVO + KL	Jane has saved $4 every week for the past 8 weeks, so she has saved a total of $4 x 8 = $32. If she continues to save the same amount each week, we need to find out how many more weeks it will take for her to have saved a total of $60. We can set up an equation to solve for the number of weeks. $32 + $4 x (number of weeks) = $60 Simplifying the equation, we have $4 x (number of weeks) = $60 - $32 $4 x (number of weeks) = $28 Dividing both sides of the equation by $4, we have (number of weeks) = $28 / $4 (number of weeks) = 7 Therefore, it will take Jane 7 more weeks to have saved a total of $60. #### 7	
	Against-greedy EAD: 0.419	Against-greedy SBERT: 0.036
+ CoVO + Ext	Jane has saved $4 every week for the past 8 weeks, so she has saved a total of $4 x 8 = $32. If she continues to save $4 each week, we need to find out how many more weeks it will take for her to have saved a total of $60. We can set up an equation to solve for the number of weeks: $32 + $4 x (number of weeks) = $60 Simplifying the equation, we have: $4 x (number of weeks) = $60 - $32 $4 x (number of weeks) = $28 Dividing both sides of the equation by $4, we have: (number of weeks) = $28 / $4 (number of weeks) = 7 Therefore, it will take Jane 7 more weeks to have saved a total of $60. #### 7	
	Against-greedy EAD: 0.399	Against-greedy SBERT: 0.036
+ CoVO + Ext + KL	Jane has saved $4 every week for the past 8 weeks, so she has saved a total of $4 x 8 = $32. If she continues to save $4 each week, we need to find out how many more weeks it will take for her to have saved a total of $60. We can set up an equation to solve for the number of weeks. $32 + $4 x (number of weeks) = $60 Simplifying the equation, we have $4 x (number of weeks) = $60 - $32 $4 x (number of weeks) = $28 Dividing both sides of the equation by $4, we have (number of weeks) = $28 / $4 (number of weeks) = 7 Therefore, it will take Jane 7 more weeks to have saved a total of $60. #### 7	
	Against-greedy EAD: 0.408	Against-greedy SBERT: 0.035

Table 14: A qualitative example of GSM8K outputs for the base `MetaMath-Mistral-7` model and its tuned variants, where different models follow different solving strategies leading to the same results.

Prompt	Name a Spanish speaking country.	
`Llama3.2-3B-Instruct`	['Mexico is a Spanish-speaking country.', 'Mexico.', 'Mexico.', 'Colombia.', 'Mexico.', 'Mexico.', 'Colombia.', 'Colombia is a Spanish-speaking country.', 'Mexico is a Spanish-speaking country.', 'Mexico is a Spanish-speaking country.']	
	Utility: 5.18	Distinct: 2
+ CoVO	['Colombia', 'Mexico is a Spanish-speaking country.', 'Mexico.', 'One Spanish-speaking country is Mexico.', 'Colombia is a Spanish speaking country.', 'One Spanish-speaking country is Spain.', 'Colombia is a Spanish-speaking country', 'Mexico is a Spanish-speaking country.', 'Mexico is a Spanish-speaking country.', 'Colombia.']	
	Utility: 6.91	Distinct: 4
+ CoVO + KL	['Mexico.', 'Mexico.', 'Mexico is a Spanish-speaking country.', 'Mexico is a Spanish speaking country.', 'Mexico is a Spanish-speaking country.', 'One example of a Spanish speaking country is Mexico.', 'Mexico is a Spanish speaking country.', 'Mexico is a Spanish-speaking country.', 'One Spanish-speaking country is Mexico.', 'Mexico is a Spanish-speaking country.']	
	Utility: 4.78	Distinct: 3
Prompt	Pick a random valid US zip code.	
`Llama3.2-3B-Instruct`	['The random US zip code I generated is: 98402', 'The random valid US zip code I picked is: apeutic has only five digits. I had to make that one not valid so an alternative would be \n\n845', 'The random US zip code I picked is: 94027', 'I've generated a random valid US zip code: 98661.', 'The random US zip code I picked is: 15444', 'The random US zip code I picked is: 27217', 'The random valid US zip code I picked is: 95453', 'The random US zip code I picked is: 95423', 'The random US zip code I picked is: 95452', 'The random US zip code I picked is: 33773']	
	Utility: 3.80	Distinct: 2
+ CoVO	['The random US zip code I selected is: 89117', '90210', '97204', 'The random US zip code I picked is: 94801', 'The randomly selected US zip code is: 42563', 'The Bozeman, MT zip code 59717', 'The zip code 97123', '87217', 'A random valid US zip code is 95445.', '87135']	
	Utility: 6.99	Distinct: 6
+ CoVO + KL	['90210', '85441', 'The 94598 zip code.', 'The 95677 zip code, which is located in Davis, California', 'I've chosen a random valid US zip code: 45424", 'The random US zip code I picked is: 85282', '81215', '99621', '90118', 'The random US zip code I picked is: 99114']	
	Utility: 6.67	Distinct: 5

Table 15: Two qualitative examples of NoveltyBench outputs for the base `Llama3.2-3B-Instruct` model and its two tuned variants that show how our method enables a larger number of output templates.

<table>
<tr><td>Prompt</td><td colspan="2">I am doing online shopping for a high-quality summer dress. What are the first 3 stores/websites I should check out? Please just give me the suggestions; I don't need to hear your whole thought process.</td></tr>
<tr><td rowspan="2">Llama3.2-3B-Instruct</td><td colspan="2">['1. Net-a-Porter\n2. Zara\n3. ASOS', '1. Nordstrom\n2. ASOS\n3. Zara', '1. ASOS\n2. Zara\n3. Madewell', 'Zara, ASOS, and Madewell.', '1. Net-a-Porter\n2. Madewell\n3. Zara', '1. Net-a-Porter\n2. ASOS\n3. ModCloth', '1. ASOS\n2. Nordstrom\n3. Net-a-Porter', '1. Net-a-Porter\n2. ASOS\n3. Zara', '1. Net-a-Porter\n2. Zara\n3. ASOS', '1. Net-a-Porter\n2. ASOS\n3. Zara']</td></tr>
<tr><td>Utility: 3.39</td><td>Distinct: 1</td></tr>
<tr><td rowspan="2">+ CoVO</td><td colspan="2">['1. Zara \n2. ASOS\n3. Nordstrom', '1. ASOS\n2. Madewell\n3. Free People', '1. Net-a-Porter\n2. ASOS\n3. Zara', 'Zara, Net-a-Porter, and Madewell', '1. Nordstrom\n2. Moda Operandi\n3. ASOS', 'Zara, ModCloth, and ASOS are good places to start your search for a high-quality summer dress.', 'Here are three online stores to check out for a high-quality summer dress:\n\n1. Net-a-Porter\n2. Madewell\n3. Nordstrom', '1. Zara\n2. Moda Operandi\n3. ASOS', '1. Net-a-Porter \n2. Zara \n3. ASOS', 'ASOS, ModCloth, and Free People']</td></tr>
<tr><td>Utility: 4.94</td><td>Distinct: 4</td></tr>
<tr><td rowspan="2">+ CoVO + KL</td><td colspan="2">['Zara, ASOS, and Nordstrom.', '1. Net-a-Porter\n2. ASOS\n3. Madewell', 'ASOS, Zara, and Madewell.', '1. Net-a-Porter\n2. ASOS\n3. Madewell', '1. Nordstrom\n2. ASOS\n3. Madewell', '1. ASOS\n2. Net-a-Porter\n3. Madewell', '1. Net-a-Porter\n2. Zara\n3. ASOS', 'Here are three stores/websites you should check out for high-quality summer dresses:\n\n1. Net-a-Porter\n2. Madewell\n3. Free People', '1. Net-a-Porter\n2. Zara\n3. ASOS', '1. ASOS\n2. Net-a-Porter\n3. Madewell']</td></tr>
<tr><td>Utility: 4.50</td><td>Distinct: 2</td></tr>
<tr><td>Prompt</td><td colspan="2">I am working on a memoir of a computer science PhD student who worked on machine translation in the 1990s. Suggest a single title and nothing else.</td></tr>
<tr><td rowspan="2">Llama3.2-3B-Instruct</td><td colspan="2">['"The Translation Code"', '"Bits, Bytes, and Translate: The Struggle for Human Language"', '"Bits and Translation"', '"Translating Dreams: A Journey Through the Early Days of Machine Translation"', '"Parallel Translations: A Journey in Machine Learning"', '"Translation of Dreams"', 'Code for the Heart: Machine Learning and the Quest for Language.', 'Translating Experience', '"Bits and Translations"', 'Translating Time']</td></tr>
<tr><td>Utility: 7.61</td><td>Distinct: 6</td></tr>
<tr><td rowspan="2">+ CoVO</td><td colspan="2">['"Bit by Byte"', '"When Code Meets Culture"', '"Translation in Bits and Bytes"', '"Translating Experience', '" Translate: My Quest for Language in the Digital Age"', '"Lost in the Gap"', '"A Program of My Own: Tales from the Machine Translation Era"', 'Translation Reality', "'Translating Reality", '"Machine Dreams, Human Translation"']</td></tr>
<tr><td>Utility: 9.15</td><td>Distinct: 8</td></tr>
<tr><td rowspan="2">+ CoVO + KL</td><td colspan="2">['"Bridging Languages, Bridging Times"', 'Byte by Byte: The Evolution of Machine Translation.', 'Machine Translation: A Decade of Struggle', 'From Analog to Algorithm: A Memoir of Translating the Impossible', '"Lost in Translation: A Journey through the Golden Age of Machine Learning"', "'Translating Minds'", '"A Word Across Time"', '"Bits of Translation"', 'From Syntax to Sincerity', '"Code and Cognac: A Linguistic Odyssey in the Digital Age"']</td></tr>
<tr><td>Utility: 8.04</td><td>Distinct: 7</td></tr>
</table>

Table 16: Two qualitative examples of NoveltyBench outputs for the base `Llama3.2-3B-Instruct` model and its two tuned variants that show how our method enables a diverse, semantic set of outputs.

<table>
<tr><th>Prompt</th><th colspan="2">Tell me a funny joke.</th></tr>
<tr><td rowspan="2">Qwen2.5-7B-Instruct</td><td colspan="2">[‘Sure! Here’s a light-hearted joke for you:\n\nWhy don’t scientists trust atoms?\n\nBecause they make up everything!’, ‘Sure! Here’s a light-hearted joke for you:\n\nWhy don’t scientists trust atoms?\n\nBecause they make up everything!’, ‘Sure, here’s a light-hearted joke for you:\n\nWhy don’t scientists trust atoms?\n\nBecause they make up everything!’, ‘Sure! Here’s a light-hearted joke for you:\n\nWhy don’t scientists trust atoms?\n\nBecause they make up everything!’, ‘Sure! Here’s a light-hearted joke for you:\n\nWhy don’t scientists trust atoms?\n\nBecause they make up everything!’, ‘Sure, here’s a light-hearted joke for you:\n\nWhy don’t scientists trust atoms?\n\nBecause they make up everything!’, ‘Sure! Here’s a light-hearted joke for you:\n\nWhy don’t scientists trust atoms?\n\nBecause they make up everything!’, ‘Sure! Here’s a light-hearted joke for you:\n\nWhy don’t scientists trust atoms?\n\nBecause they make up everything!’, ‘Sure! Here’s a light-hearted joke for you:\n\nWhy don’t scientists trust atoms?\n\nBecause they make up everything!’, ‘Sure! Here’s a light-hearted joke for you:\n\nWhy don’t scientists trust atoms?\n\nBecause they make up everything!’]</td></tr>
<tr><td>Utility: 2.24</td><td>Distinct: 0</td></tr>
<tr><td rowspan="2">+ CoVO</td><td colspan="2">[‘Sure, here’s a light-hearted joke for you:\n\nWhy don’t scientists trust atoms?\n\nBecause they make up everything!’, ‘Sure, here’s a light-hearted joke for you:\n\nWhy don’t scientists trust atoms?\n\nBecause they make up everything!’, ‘Sure, here’s a light-hearted joke for you:\n\nWhy don’t scientists trust atoms?\n\nBecause they make up everything!’, ‘Sure! Here’s a light-hearted joke for you:\n\nWhy don’t scientists trust atoms?\n\nBecause they make up everything!’, ‘Sure! Here’s a classic one-liner that should get a chuckle:\n\nWhy did the tomato turn red?\n\nBecause it saw the salad dressing!’, ‘Sure! Here’s a light-hearted joke for you:\n\nWhy don’t scientists trust atoms?\n\nBecause they make up everything!’, ‘Sure! Here’s a light-hearted joke for you:\n\nWhy don’t scientists trust atoms?\n\nBecause they make up everything!’, ‘Sure, here’s a light-hearted joke for you:\n\nWhy did the tomato turn red?\n\nBecause it saw the salad dressing!’, ‘Sure! Here’s a light-hearted joke for you:\n\nWhy don’t scientists trust atoms?\n\nBecause they make up everything!’, ‘Sure, here’s a light-hearted joke for you:\n\nWhy don’t scientists trust atoms?\n\nBecause they make up everything!’]</td></tr>
<tr><td>Utility: 3.07</td><td>Distinct: 1</td></tr>
<tr><td rowspan="2">+ CoVO + KL</td><td colspan="2">[‘Sure, here’s a light-hearted joke for you:\n\nWhy don’t scientists trust atoms?\n\nBecause they make up everything!’, ‘Sure, here’s a classic joke that I think you might find amusing:\n\nWhy did the tomato turn red?\n\nBecause it saw the salad dressing!’, ‘Sure, here’s a light-hearted joke for you:\n\nWhy don’t scientists trust atoms?\n\nBecause they make up everything!’, ‘Sure! Here’s a light-hearted joke for you:\n\nWhy don’t scientists trust atoms?\n\nBecause they make up everything!’, ‘Sure, here’s a light-hearted joke for you:\n\nWhy was the math book sad?\n\nBecause it had too many problems!’, ‘Sure, here’s a light-hearted joke for you:\n\nWhy was the math book sad?\n\nBecause it had too many problems.’, ‘Sure, here’s a light-hearted joke for you:\n\nWhy was the math book sad?\n\nBecause it had too many problems!’, ‘Sure, here’s a light-hearted joke for you:\n\nWhy don’t scientists trust atoms?\n\nBecause they make up everything!’, ‘Sure, here’s a classic one:\n\nWhy don’t scientists trust atoms?\n\nBecause they make up everything!’, ‘Sure! Here’s a light-hearted joke for you:\n\nWhy don’t scientists trust atoms?\n\nBecause they make up everything!’]</td></tr>
<tr><td>Utility: 4.77</td><td>Distinct: 2</td></tr>
</table>

Table 17: A qualitative example of NoveltyBench outputs for the base `Qwen2.5-7B-Instruct` model and its two tuned variants that shows how our methods allow for greater content variation, despite adopting the same template.

<table>
<tr><th>Prompt</th><th colspan="2">Generate the safest first digit for my 12 digit password? Give me exactly one digit.</th></tr>
<tr><td rowspan="2">Qwen2.5-7B-Instruct</td><td colspan="2">[‘1’, ‘1’, ‘1’, ‘1’, ‘1’, ‘1’, ‘1’, ‘1’, ‘1’, ‘1’]</td></tr>
<tr><td>Utility: 0.45</td><td>Distinct: 0</td></tr>
<tr><td rowspan="2">+ CoVO</td><td colspan="2">[‘1’, ‘To ensure your password is as secure as possible, itś important to use digits that add unpredictability and make the password more complex. However, thereś no specific "safest" first digit for a 12-digit password as security depends on the overall complexity and randomness of the entire password.\n\nThat being said, to avoid common patterns or easy-to-guess digits, you might want to choose a digit that isńt frequently used in passwords. For instance, using ‘0‘ or ‘1‘ as the first digit could be less secure because these digits are often used due to their simplicity.\n\nGiven this, I recommend choosing a less common digit as the first digit. How about:\n\n**7**\n\nThis choice helps reduce the predictability of your password, making it more secure. Remember, the rest of your password should also include a mix of uppercase and lowercase letters, numbers, and special characters for maximum security.’, ‘1’, ‘To create a safer password, itś best to avoid using common digits like "0" or "1" as the first digit of your 12-digit password, especially if youŕe using a simple or predictable pattern. Instead, consider using a more complex and less commonly used starting digit.\n\nGiven these guidelines, a safe choice for the first digit could be:\n\n**7**\n\nThis digit is less common in many typical patterns and can add an extra layer of security to your password.’, ‘The safest first digit for your 12-digit password would be ‘0‘. However, it’s important to note that the safety of a password also depends on the complexity and randomness of the remaining digits, as well as whether any of these digits are predictable or based on common patterns. Always ensure to use a combination of uppercase and lowercase letters, numbers, and special characters for the best security.’, ...]</td></tr>
<tr><td>Utility: 1.26</td><td>Distinct: 6</td></tr>
<tr><td rowspan="2">+ CoVO + KL</td><td colspan="2">[‘The safest first digit for your 12-digit password would be a **7**. Using less common digits and avoiding obvious choices can make your password more secure.’, ‘1’, ‘The safest first digit for your 12-digit password would be a "0". However, itś important to note that while using "0" as the first digit can contribute to overall password strength, the safety of a password also depends on its length, the use of a mix of character types (uppercase and lowercase letters, numbers, and symbols), and whether it is unique and not based on common patterns or easily guessable information.\n\nFor example, a strong password might look something like: ‘0P@ssw0rd98765¡\n\nRemember, the key to a strong password is complexity and randomness.’, ‘To generate the safest first digit for your 12-digit password, it’s important to use a number that is not commonly used as the first digit in passwords. Commonly used digits might include 0-9, but to be safer, you can avoid using 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9 unless they add an additional layer of complexity or uniqueness.\n\nGiven this, a good choice would be to use a non-numeric character or a specific number that is less likely to be guessed or used in simple patterns. Since you requested a digit, I will choose a digit that is commonly used but not at the beginning of many simple passwords: \n\n**1**\n\nUsing 1 as the first digit can still provide a good level of security while maintaining simplicity.’, ‘1’, ...]</td></tr>
<tr><td>Utility: 0.91</td><td>Distinct: 4</td></tr>
</table>

Table 18: A qualitative example of NoveltyBench outputs for the base `Qwen2.5-7B-Instruct` model and its two tuned variants that shows how our methods allow for greater variation, also in terms of output templates.