Availability of DSR-REX, Baselines and Dataset Please find our code repository at:

```
https://anonymous.4open.science/status/dsr-rex-1876
```

1. the implementation of our DSR-REX method is in the folder "`dsr_rex_pytorch/`".
2. the list of datasets is listed in "`data_oracle/scibench/scibench/data/`".
3. the implementation of several baseline algorithms is collected in folder "`baslines/`".

We provide a "README.md" document for executing the programs.

We summarize the supplementary material as follows: Section A details the used mathematical laws for deriving symbolic forms of expressions. Section B and section C give a detailed theoretical explanation of the proposed method. Section D details the experimental settings.

A Implementation of Symbolic Reasoning Module

We consider a wide list of mathematical properties.

1. Commuative law, for example, $a + b = b + a$ or $a * b = b * a$.
2. Distributive laws, for example $(x + y)^2 = x^2 + 2xy + y^2$.
3. Factorize an expression into simpler components. For example, $x^2 - y^2 = (x - y)(x + y)$.
4. Sum-to-Product Identities:
$$\sin(a) + \sin(b) = 2\sin\left(\frac{a+b}{2}\right), \cos\left(\frac{a-b}{2}\right)$$
$$\cos(a) + \cos(b) = 2\cos\left(\frac{a+b}{2}\right)\cos\left(\frac{a-b}{2}\right)$$
5. Product-to-Sum Identities:
$$\sin(a)\cos(b) = \frac{1}{2}[\sin(a+b) + \sin(a-b)]$$
$$\cos(a)\cos(b) = \frac{1}{2}[\cos(a+b) + \cos(a-b)]$$
$$\sin(a)\sin(b) = \frac{1}{2}[\cos(a-b)\cos(a+b)]$$
6. Double Angle Formulas:
$$\sin(2a) = 2\sin(a)\cos(a), \quad \cos(2a) = \cos^2(a)\sin^2(a)$$
7. Co-function Identities:
$$\sin\left(\frac{\pi}{2} - x\right) = \cos(x), \quad \cos\left(\frac{\pi}{2} - x\right) = \sin(x),$$
$$\tan\left(\frac{\pi}{2} - x\right) = \cot(x), \quad \cot\left(\frac{\pi}{2} - x\right) = \tan(x)$$
8. Pythagorean Identities:
$$\sin^2(x) + \cos^2(x) = 1, \qquad 1 + \tan^2(x) = \sec^2(x),$$
$$1 + \cot^2(x) = \csc^2(x)$$
9. Half-Angle Formulas:
$$\sin\left(\frac{x}{2}\right) = \pm\sqrt{\frac{1-\cos(x)}{2}}, \qquad \cos\left(\frac{x}{2}\right) = \pm\sqrt{\frac{1+\cos(x)}{2}},$$
$$\tan\left(\frac{x}{2}\right) = \pm\sqrt{\frac{1-\cos(x)}{1+\cos(x)}}.$$

Rule	expression
Product-to-Sum Identities	$\sin(a) + \sin(b) = 2\sin\left(\frac{a+b}{2}\right)\cos\left(\frac{a-b}{2}\right)$ $\cos(a) + \cos(b) = 2\cos\left(\frac{a+b}{2}\right)\cos\left(\frac{a-b}{2}\right)$
Product-to-Sum Identities	$\sin(a)\cos(b) = \frac{1}{2}\left[\sin(a+b) + \sin(a-b)\right]$ $\cos(a)\cos(b) = \frac{1}{2}\left[\cos(a+b) + \cos(a-b)\right]$ $\sin(a)\sin(b) = \frac{1}{2}\left[\cos(a-b) - \cos(a+b)\right]$
Double Angle Formulas	$\sin(2a) = 2\sin(a)\cos(a)$ $\cos(2a) = \cos^2(a) - \sin^2(a)$ $\cos(2a) = 2\cos^2(a) - 1$ $\cos(2a) = 1 - 2\sin^2(a)$ $\tan(2a) = \frac{2\tan(a)}{1-\tan^2(a)}$
Co-function Identities	$\sin\left(\frac{\pi}{2} - a\right) = \cos(a)$ $\cos\left(\frac{\pi}{2} - a\right) = \sin(a)$ $\tan\left(\frac{\pi}{2} - a\right) = \cot(a)$ $\cot\left(\frac{\pi}{2} - a\right) = \tan(a)$
Pythagorean Identities	$\sin^2(a) + \cos^2(a) = 1$ $1 + \tan^2(a) = \sec^2(a)$ $1 + \cot^2(a) = \csc^2(a)$
Half-Angle Formulas	$\sin\left(\frac{a}{2}\right) = \pm\sqrt{\frac{1-\cos(a)}{2}}$ $\cos\left(\frac{a}{2}\right) = \pm\sqrt{\frac{1+\cos(a)}{2}}$ $\tan\left(\frac{a}{2}\right) = \pm\sqrt{\frac{1-\cos(a)}{1+\cos(a)}}$
Sum and Difference Formulas	$\sin(a \pm b) = \sin(a)\cos(b) \pm \cos(a)\sin(b)$ $\cos(a \pm b) = \cos(a)\cos(b) \mp \sin(a)\sin(b)$ $\tan(a \pm b) = \frac{\tan(a)\pm\tan(b)}{1\mp\tan(a)\tan(b)}$

Table 2: Mathematical Identities

10. Sum and Difference Formulas

$$\sin(a \pm b) = \sin(a)\cos(b) \pm \cos(a)\sin(b),$$
$$\cos(a \pm b) = \cos(a)\cos(b) \mp \sin(a)\sin(b),$$
$$\tan(a \pm b) = \frac{\tan(a) \pm \tan(b)}{1 \mp \tan(a)\tan(b)}$$

11. Double-Angle Formulas:

$$\sin(2a) = 2\sin(a)\cos(a), \qquad \cos(2a) = \cos^2(a)\sin^2(a),$$
$$\cos(2a) = \cos^2(a) - \sin^2(a) \qquad \tan(2a) = \frac{2\tan(a)}{1 - \tan^2(a)}$$

12. exp and log rules:

$$\exp(a + b) = \exp(a) \cdot \exp(b) \qquad \log(ab) = \log(a) + \log(b)$$

B Proof of Theorem 1

Theorem. **(1)** The expectation of reward over probability distribution $p_\theta(\tau)$ equals the expectation over probability distribution $q_\theta(\phi)$, that is:

$$\mu = \mathbb{E}_{\tau \sim p_\theta}[R(\tau)] = \mathbb{E}_{\phi \sim q_\theta}[R(\phi)].$$

(2) The expectation of **policy gradient** over probability distribution $p_\theta(\tau)$ equals the expectation over probability distribution $q_\theta(\phi)$, that is:

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau \sim p_\theta}[R(\tau)\nabla_\theta \log p_\theta(\tau)] = \mathbb{E}_{\phi \sim q_\theta}[R(\phi)\nabla_\theta \log q_\theta(\phi)].$$

Symbol	Definition
τ	a sequence of math operators, variables, and coefficients
Π	the set of all sequences
$\mathcal{S}_\phi \subseteq \Pi$	the subset of sequences which can be constructed into the same expression ϕ
ϕ	an expression
Φ	the set of all expressions
$\phi = \texttt{MAP}(\tau)$	map the sequence τ to its corresponding expression ϕ
$p_\theta(\tau)$	the probability of sampling sequence
$q_\theta(\phi)$	the probability of sampling expression ϕ
$g_\theta(\tau)$	$R(\tau)\nabla_\theta \log p_\theta(\tau)$

Table 3: List of notations used in this work.

It states the new objective as defined in Equation 5 is the same as the classic objective (defined in Equation 1).

Proof. Define Π as the set of all possible sequences and Φ as the set of all possible expressions. We denote $\mathcal{S}_\phi = \{\tau | \texttt{MAP}(\tau) = \phi\}$ as the group of sequences that can be constructed into expression ϕ.
Part 1. The expectation can be expanded as:

$$\begin{aligned}\mu = \mathbb{E}_{\phi\sim q_\theta}[R(\phi)] = \sum_{\phi\in\Phi} R(\phi)q_\theta(\phi) = \sum_{\phi\in\Phi} R(\phi) \sum_{\tau\in\mathcal{S}_\phi} p_\theta(\tau) &= \sum_{\phi\in\Phi}\sum_{\tau\in\mathcal{S}_\phi} R(\tau)p_\theta(\tau) \\ &= \sum_{\tau\in\Pi} R(\tau)p_\theta(\tau) = \mathbb{E}_{\tau\sim p_\theta}[R(\tau)]\end{aligned}$$

The first and second equalities are due to the definitions of expectation and q_θ. The third equality is obtained by observing the fact all sequences τ in group $\mathcal{S}_\phi$ share the same reward value $R(\tau) = R(\tau')$ for $\tau, \tau' \in \mathcal{S}_\phi$. The fourth equality is obtained because partitioning all trajectories by groups and then summing over each trajectory in the group is the same as summing over all trajectories directly.

Part 2. Based on the definition of expectation, the left-hand side and the right-hand side of policy gradient:

$$\mathbb{E}_{\phi\sim q_\theta}[R(\phi)\nabla_\theta \log q_\theta(\phi)] = \sum_{\phi\in\Phi} \left(R(\phi)\nabla_\theta \log q_\theta(\phi)\right) q_\theta(\phi) = \sum_{\phi\in\Phi} R(\phi)\nabla_\theta q_\theta(\phi)$$

$$\mathbb{E}_{\tau\sim p_\theta}[R(\tau)\nabla_\theta \log p_\theta(\tau)] = \sum_{\tau\in\Pi} \left(R(\tau)\nabla_\theta \log p_\theta(\tau)\right) p_\theta(\tau) = \sum_{\tau\in\Pi} R(\tau)\nabla_\theta p_\theta(\tau)$$

The problem is transformed into showing if the derived summation equals to each other. By the definition of $q_\theta(\phi)$, we have:

$$\nabla_\theta q_\theta(\phi) = \nabla_\theta \sum_{\tau\in\mathcal{S}_\phi} p_\theta(\tau) = \sum_{\tau\in\mathcal{S}_\phi} \nabla_\theta p_\theta(\tau)$$

The last step holds since the gradient operator is linear to the summation operator. Then we have:

$$\sum_{\phi\in\Phi} R(\phi)\nabla_\theta q_\theta(\phi) = \sum_{\phi\in\Phi} R(\phi) \sum_{\tau\in\mathcal{S}_\phi} \nabla_\theta p_\theta(\tau) = \sum_{\phi\in\Phi}\sum_{\tau\in\mathcal{S}_\phi} R(\tau)\nabla_\theta p_\theta(\tau) = \sum_{\tau\in\Pi} R(\tau)\nabla_\theta p_\theta(\tau)$$

The last step holds because partitioning all sequences by groups and then summing over each sequence in the group is the same as summing over all sequences directly. Putting it all together, we have:

$$\mathbb{E}_{\phi\sim q_\theta}[R(\phi)\nabla_\theta \log q_\theta(\phi)] = \mathbb{E}_{\tau\sim p_\theta}[R(\tau)\nabla_\theta \log p_\theta(\tau)].$$

This completes the proof. □

One important conclusion from the above theorem that will be useful for the following proof is $\sum_{\phi\in\Phi} R(\phi)q_\theta(\phi) = \sum_{\tau\in\Pi} R(\tau)p_\theta(\tau)$. It implies:

$$\sum_{\phi\in\Phi} R^2(\phi)q_\theta(\phi) = \sum_{\tau\in\Pi} R^2(\tau)p_\theta(\tau)$$

Lemma 1. The variance of the **policy gradient** over probability distribution $p_\theta(\tau)$ is larger than the variance over probability distribution $q_\theta(\phi)$, that is:

$$\mathbb{V}\mathrm{ar}_{\tau\sim p_\theta}[R(\tau)\nabla_\theta \log p_\theta(\tau)] \geq \mathbb{V}\mathrm{ar}_{\phi\sim q_\theta}[R(\phi)\nabla_\theta \log q_\theta(\phi)].$$

Proof. By the definition of variance, we have:

$$\begin{aligned}\mathbb{V}\mathrm{ar}_{\tau\sim p_\theta}[R(\tau)\nabla_\theta \log p_\theta(\tau)] &= \sum_{\tau'\in\Pi}(R(\tau')\nabla_\theta \log p_\theta(\tau') - \nabla_\theta J(\theta))^2 p_\theta(\tau')\\ &= \sum_{\tau'\in\Pi} R^2(\tau')\frac{(\nabla_\theta p_\theta(\tau'))^2}{p_\theta(\tau')} - 2\nabla_\theta J(\theta)\sum_{\tau'\in\Pi} R(\tau')\nabla_\theta p_\theta(\tau') + (\nabla_\theta J(\theta))^2 \sum_{\tau'\in\Pi} p_\theta(\tau')\end{aligned}$$

where $\nabla_\theta J(\theta)$ indicates the expectation of policy gradient and is introduced in Theorem 1 and the second row is obtained by expanding the whole equation. By Theorem 1, the first part can be lower bounded as:

$$\sum_{\tau'\in\Pi} R^2(\tau')\frac{(\nabla_\theta p_\theta(\tau'))^2}{p_\theta(\tau')} \geq \sum_{\phi'\in\Phi} R^2(\phi')\frac{(\nabla_\theta q_\theta(\phi'))^2}{q_\theta(\phi')}$$

By Theorem 1, the second part equals to:

$$-2\nabla_\theta J(\theta)\sum_{\tau'\in\Pi} R(\tau')\nabla_\theta p_\theta(\tau') = -2\nabla_\theta J(\theta)\sum_{\phi'\in\Phi} R(\phi')\nabla_\theta q_\theta(\phi')$$

Since the probability mass equals one, i.e., $\sum_{\phi'\in\Phi} q_\theta(\phi') = \sum_{\tau'\in\Pi} p_\theta(\tau') = 1$, the third parts equals to:

$$(\nabla_\theta J(\theta))^2 \sum_{\tau'\in\Pi} p_\theta(\tau') = (\nabla_\theta J(\theta))^2 \sum_{\phi'\in\Phi} q_\theta(\phi')$$

To conclude, we have:

$$\begin{aligned}&\mathbb{V}\mathrm{ar}_{\tau\sim p_\theta}[R(\tau)\nabla_\theta \log p_\theta(\tau)]\\ &\quad\geq \sum_{\phi'\in\Phi} R^2(\phi')\frac{(\nabla_\theta q_\theta(\phi'))^2}{q_\theta(\phi')} - 2\nabla_\theta J(\theta)\sum_{\phi'\in\Phi} R(\phi')\nabla_\theta q_\theta(\phi') + (\nabla_\theta J(\theta))^2 \sum_{\phi'\in\Phi} q_\theta(\phi')\\ &\quad= \sum_{\phi'\in\Phi}(R(\phi')\nabla_\theta \log q_\theta(\phi') - \nabla_\theta J(\theta))^2 q_\theta(\phi')\\ &\quad= \mathbb{V}\mathrm{ar}_{\phi\sim q_\theta}[R(\phi)\nabla_\theta \log q_\theta(\phi)]\end{aligned}$$

To conclude, we obtain the final result:

$$\mathbb{V}\mathrm{ar}_{\tau\sim p_\theta}[R(\tau)\nabla_\theta \log p_\theta(\tau)] \geq \mathbb{V}\mathrm{ar}_{\phi\sim q_\theta}[R(\phi)\nabla_\theta \log q_\theta(\phi)].$$

This completes the proof. □

C Proof of Theorem 2

Theorem. Using N samples $\tau_1, \ldots, \tau_N$ from distribution p_θ together with an reasoning engine to obtain extra samples. **(1)** Unbiased Estimator. The expectation of $\widehat{\nabla}_\theta J(\theta)$ over distribution $p_\theta(\tau)$ equals to $\nabla_\theta J(\theta)$ that is:

$$\mathbb{E}_{\tau\sim p_\theta}\left[\widehat{\nabla}_\theta J(\theta)\right] = \mathbb{E}_{\phi\sim q_\theta}[R(\phi)\nabla_\theta \log q_\theta(\phi)] = \nabla_\theta J(\theta).$$

(2) Variance Reduction. The variance of the proposed estimator is smaller than $\widetilde{\nabla}_\theta J(\theta)$:

$$\mathbb{V}\mathrm{ar}_{\phi\sim q_\theta}\left[\widehat{\nabla}_\theta J(\phi)\right] \leq \mathbb{V}\mathrm{ar}_{\tau\sim p_\theta}\left[\widetilde{\nabla}_\theta J(\theta)\right]$$

Proof. **Proof of Unbiased Estimator.** If we sample $\tau_1, \dots, \tau_N$ from distribution p_θ, the proposed estimator is defined as follows:

$$\widehat{\nabla}_\theta J(\theta) = \frac{1}{N} \sum_{i=1}^{N} \sum_{\phi \in \Phi} \mathbb{I}\{\texttt{MAP}(\tau_i) = \phi\} R(\phi) \nabla_\theta \log q_\theta(\phi)$$

By adding the expectation over the distribution $p_\theta(\tau)$, we have:

$$\begin{aligned}\mathbb{E}_{\tau \sim p_\theta}\left[\widehat{\nabla}_\theta J(\theta)\right] &= \mathbb{E}_{\tau \sim p_\theta}\left[\frac{1}{N} \sum_{i=1}^{N} \sum_{\phi \in \Phi} \mathbb{I}\{\texttt{MAP}(\tau_i) = \phi\} R(\phi) \nabla_\theta \log q_\theta(\phi)\right] \\ &= \mathbb{E}_{\tau \sim p_\theta}\left[\sum_{\phi \in \Phi} \mathbb{I}\{\texttt{MAP}(\tau) = \phi\} R(\phi) \nabla_\theta \log q_\theta(\phi)\right]\end{aligned}$$

Where the second row is obtained by the expectation operator is linear and the samples are drawn i.i.d. from the distribution. We then expand the right-hand side with the definition of expectation:

$$\begin{aligned}\mathbb{E}_{\tau \sim p_\theta}\left[\widehat{\nabla}_\theta J(\theta)\right] &= \sum_{\tau \in \Pi}\left[\sum_{\phi \in \Phi} \mathbb{I}\{\texttt{MAP}(\tau) = \phi\} R(\phi) \nabla_\theta \log q_\theta(\phi)\right] p_\theta(\tau) \\ &= \sum_{\phi \in \Phi} \underbrace{\sum_{\tau \in \Pi} \mathbb{I}\{\texttt{MAP}(\tau) = \phi\} p_\theta(\tau)}_{\text{by the definition in Equation 4}} R(\phi) \nabla_\theta \log q_\theta(\phi) \\ &= \sum_{\phi \in \Phi} q_\theta(\phi) R(\phi) \nabla_\theta \log q_\theta(\phi) \\ &= \mathbb{E}_{\phi \sim q_\theta}[R(\phi) \nabla_\theta \log q_\theta(\phi)]\end{aligned}$$

where the last line is obtained by Theorem 1. In practice, we only have one sequence decoder that models over sequences with probability distribution p_θ. The lemma states that using sampled sequences $\tau_1, \dots, \tau_N$ from the distribution p_θ is the same as drawing sampled expressions from the distribution q_θ. Based on Theorem 1, we show that the gradient estimator is unbiased:

$$\mathbb{E}_{\tau \sim p(\tau)}\left[\widehat{\nabla}_\theta J(\theta)\right] = \nabla_\theta J(\theta).$$

The above steps justify that the proposed expression reasoning module helps to compute the correct gradient values (i.e., unbiased gradient estimator).

Proof of Variance Reduction. The sample variance of the original policy gradient is defined as:

$$\mathbb{V}\mathrm{ar}_{\tau \sim p_\theta}\left[\widetilde{\nabla}_\theta J(\theta)\right] = \mathbb{V}\mathrm{ar}_{\tau \sim p_\theta}\left[\frac{1}{N} \sum_{i=1}^{N} R(\tau_i) \nabla_\theta \log p_\theta(\tau_i)\right] = \frac{1}{N} \mathbb{V}\mathrm{ar}_{\tau \sim p_\theta}\left[R(\tau) \nabla_\theta \log p_\theta(\tau)\right]$$

The sampled variance of the proposed method is:

$$\begin{aligned}\mathbb{V}\mathrm{ar}_{\tau \sim p_\theta}\left[\widehat{\nabla}_\theta J(\theta)\right] &= \mathbb{V}\mathrm{ar}_{\tau \sim p_\theta}\left[\frac{1}{N} \sum_{i=1}^{N} \sum_{\phi \in \Phi} \mathbb{I}\{\texttt{MAP}(\tau_i) = \phi\} R(\phi) \nabla_\theta \log q_\theta(\phi)\right] \\ &= \frac{1}{N} \mathbb{V}\mathrm{ar}_{\tau \sim p_\theta}\left[\sum_{\phi \in \Phi} \mathbb{I}\{\texttt{MAP}(\tau) = \phi\} R(\phi) \nabla_\theta \log q_\theta(\phi)\right]\end{aligned}$$

Using the result in part 1 where we show the mean of the quantity is $\nabla_\theta J(\theta)$, we can expand by the definition of variance:

$$
\begin{aligned}
\mathbb{V}\mathrm{ar}_{\tau\sim p_\theta}\left[\widehat{\nabla}_\theta J(\theta)\right] &= \sum_{\tau'\in\Pi}\left(\sum_{\phi\in\Phi}\mathbb{I}\{\texttt{MAP}(\tau')=\phi\}R(\phi)\nabla_\theta\log q_\theta(\phi) - \nabla_\theta J(\theta)\right)^2 p_\theta(\tau') \\
&= \sum_{\phi\in\Phi}\underbrace{\sum_{\tau'\in\Pi}\mathbb{I}\{\texttt{MAP}(\tau')=\phi\}p_\theta(\tau')}_{\text{By the definition in Equation 4}}\left(R(\tau')\nabla_\theta\log q_\theta(\phi)\right)^2 \\
&\quad -2\nabla_\theta J(\theta)\sum_{\phi\in\Phi}\underbrace{\sum_{\tau'\in\Pi}\mathbb{I}\{\texttt{MAP}(\tau)=\phi\}p_\theta(\tau')}_{\text{By the definition in Equation 4}}R(\tau')\nabla_\theta\log q_\theta(\phi) \\
&\quad +\left(\nabla_\theta J(\theta)\right)^2\sum_{\tau'\in\Pi}p_\theta(\tau')
\end{aligned}
$$

After plugin in Equation 4, we can have:

$$
\begin{aligned}
&= \sum_{\phi\in\Phi}q_\theta(\phi)\left(R(\tau')\nabla_\theta\log q_\theta(\phi)\right)^2 - 2\nabla_\theta J(\theta)\sum_{\phi\in\Phi}q_\theta(\phi)R(\tau')\nabla_\theta\log q_\theta(\phi) + \left(\nabla_\theta J(\theta)\right)^2\sum_{\phi\in\Phi}q_\theta(\phi) \\
&= \sum_{\phi\in\Phi}(R(\phi)\nabla_\theta\log q_\theta(\phi) - \nabla_\theta J(\theta))^2 q_\theta(\phi) \\
&= \frac{1}{N}\mathbb{V}\mathrm{ar}_{\phi\sim q_\theta}\left[R(\phi)\nabla_\theta\log q_\theta(\phi)\right].
\end{aligned}
$$

Due to the linearity of summation, we obtain the second row by switching the two summations, i.e., $\sum_{\phi\in\Phi}$ and $\sum_{\tau'\in\Pi}$. Based on Lemma 1, we can conclude that the proposed sampler enjoys a smaller variance:

$$\mathbb{V}\mathrm{ar}_{\phi\sim q_\theta}[\widehat{\nabla}_\theta J(\theta)] \leq \mathbb{V}\mathrm{ar}_{\tau\sim p_\theta}[\widetilde{\nabla}_\theta J(\theta)].$$

This completes the proof. □

Lemma 2. $\sum_{\tau'\in\Pi} R^2(\tau')\frac{(\nabla_\theta p_\theta(\tau'))^2}{p_\theta(\tau')} \geq \sum_{\phi\in\Phi} R^2(\phi')\frac{(\nabla_\theta q_\theta(\phi'))^2}{q_\theta(\phi')}$.

Proof. The first step is to rewrite the left-hand side by changing the summation over sequences to summation over groups of sequences with the same reward,

$$\sum_{\tau'\in\Pi}R^2(\tau')\frac{(\nabla_\theta p_\theta(\tau'))^2}{p_\theta(\tau')} = \sum_{\phi\in\Phi}\sum_{\tau'\in\mathcal{S}_\phi}R^2(\tau')\frac{(\nabla_\theta p_\theta(\tau'))^2}{p_\theta(\tau')} = \sum_{\phi\in\Phi}R^2(\phi)\sum_{\tau'\in\mathcal{S}_\phi}\frac{(\nabla_\theta p_\theta(\tau'))^2}{p_\theta(\tau')}$$

Then we show, in each group of sequences $\mathcal{S}_\phi$, the left-hand side is larger:

$$\left(\sum_{\tau'\in\mathcal{S}_\phi}\frac{(\nabla_\theta p_\theta(\tau'))^2}{p_\theta(\tau')}\right) - \frac{(\nabla_\theta q_\theta(\phi'))^2}{q_\theta(\phi')} = \left(\sum_{\tau'\in\mathcal{S}_\phi}\frac{(\nabla_\theta p_\theta(\tau'))^2}{p_\theta(\tau')}\right) - \frac{\left(\sum_{\tau\in\mathcal{S}_\phi}\nabla_\theta p_\theta(\tau)\right)^2}{\sum_{\tau\in\mathcal{S}_\phi}p_\theta(\tau)} > 0$$

The last inequality is obtained from Lemma 4 by relating scalar $p_\theta(\tau)$ with b_i and vector/matrix $\nabla_\theta p_\theta(\tau)$ with $\mathbf{a}_i$.Therefore, the final result holds. □

Lemma 3 (log-derivative trick)**.** Let $p_\theta(\tau)\in(0,1)$ represents a probability distribution over input τ with parameters θ and notation ∇_θ is the partial derivative with respect to θ.

$$\nabla_\theta p_\theta(\tau) = p_\theta(\tau)\nabla_\theta\log p_\theta(\tau).$$

Lemma 4. For any real-valued vector $\mathbf{a}_i\in\mathbb{R}^d$ and positive real numbers b_i, for $i=1\ldots,n$. We have:

$$\sum_{i=1}^{n}\frac{\mathbf{a}_i^\top\mathbf{a}_i}{b_i} - \frac{\left(\sum_{i=1}^n\mathbf{a}_i\right)^\top\left(\sum_{i=1}^n\mathbf{a}_i\right)}{\sum_{i=1}^n b_i} \geq 0$$

Proof. The idea to prove is inspired from Sedrakyan & Sedrakyan (2018). To show that:

$$\sum_{i=1}^{n} \frac{\mathbf{a}_i^\top a_i}{b_i} - \frac{\left(\sum_{i=1}^{n} \mathbf{a}_i\right)^\top \left(\sum_{i=1}^{n} a_i\right)}{\sum_{i=1}^{n} b_i} \geq 0,$$

Let's denote $\mathbf{a}_i \in \mathbb{R}^d$ and $b_i \in \mathbb{R}_{>0}$ for $i = 1, \dots, n$. Consider vectors $\mathbf{a}_i \in \mathbb{R}^d$ and scalars $\sqrt{b_i}$ and apply the Cauchy-Schwarz inequality:

$$\begin{aligned}
\left(\sum_{i=1}^{n} \frac{\mathbf{a}_i}{\sqrt{b_i}} \cdot \sqrt{b_i}\right)^2 &\leq \left(\sum_{i=1}^{n} \left\| \frac{\mathbf{a}_i}{\sqrt{b_i}} \right\|^2\right) \left(\sum_{i=1}^{n} b_i\right) \\
\left(\sum_{i=1}^{n} \mathbf{a}_i\right)^2 &\leq \left(\sum_{i=1}^{n} \frac{\mathbf{a}_i^\top \mathbf{a}_i}{b_i}\right) \left(\sum_{i=1}^{n} b_i\right) \\
\left(\sum_{i=1}^{n} \mathbf{a}_i\right)^\top \left(\sum_{i=1}^{n} \mathbf{a}_i\right) &\leq \left(\sum_{i=1}^{n} \frac{\mathbf{a}_i^\top \mathbf{a}_i}{b_i}\right) \left(\sum_{i=1}^{n} b_i\right).
\end{aligned}$$

Rewriting the inequality, we obtain:

$$\left(\sum_{i=1}^{n} b_i\right) \left(\sum_{i=1}^{n} \frac{\mathbf{a}_i^\top \mathbf{a}_i}{b_i}\right) - \left(\sum_{i=1}^{n} \mathbf{a}_i\right)^\top \left(\sum_{i=1}^{n} \mathbf{a}_i\right) \geq 0.$$

Dividing through by $\sum_{i=1}^{n} b_i$, we get:

$$\sum_{i=1}^{n} \frac{\mathbf{a}_i^\top \mathbf{a}_i}{b_i} - \frac{\left(\sum_{i=1}^{n} \mathbf{a}_i\right)^\top \left(\sum_{i=1}^{n} \mathbf{a}_i\right)}{\sum_{i=1}^{n} b_i} \geq 0,$$

which completes the proof. □

D Experiment Settings

D.1 Hyper-parameters Configurations

When fitting the values of open constants in each expression, we sample a batch of data with batch size 1024 from the data Oracle. The open constants in the expressions are fitted on the data using the BFGS optimizer[1]. We use a multi-processor library to fit multiple expressions using 8 CPU cores in parallel. This greatly reduced the total training time.

An expression containing placeholder symbol A or containing more than 20 open constants is not evaluated on the data, the fitness score of it is $-\infty$. In terms of the reward function in the policy gradient objective, we use $\texttt{reward}(\tau) = \frac{1}{1+\texttt{NMSE}(\phi)}$. The normalized mean-squared error metric is further defined in Equation 8.

The deep network part is implemented using the most recent version of TensorFlow, the expression evaluation is based on the Sympy library, and the step for fitting open constants in expression with the dataset uses the Scipy library. We further summary all the above necessary configurations in Table 4.

D.2 Loss Benchmark of DSR-Rex

1. The vanilla policy gradient follows the definition in Equation 7.
2. we choose the baseline function as the average of the reward of the current sampled batch expressions. Thus we have:

$$\nabla_\theta J(\theta) \approx \frac{1}{N} \sum_{i=1}^{N} (R(\tau^i) - b) \nabla_\theta \log p_\theta(\tau^i), \qquad \text{where } b = \sum_{i=1}^{N} R(\tau^i)$$

[1] `https://docs.scipy.org/doc/scipy/reference/optimize.minimize-bfgs.html`

General Parameters	
max length of generated sequence	20
batch size of generated sequence	1024
total learning iterations	200
Reward function	$\texttt{reward}(\tau) = \frac{1}{1+\texttt{NMSE}(\phi)}$
Expressions and Dataset	
training dataset size	2048
validation and testing dataset size	2048
coefficient fitting optimizer	BFGS
maximum allowed coefficients	20
optimization termination criterion	error is less than $1e-6$
Deep Neural Network Optimizer	
optimizer	Adam
learning rate	0.009
entropy weight	0.03
entropy gamma	0.7

Table 4: Hyper-parameter configurations.

3. policy gradient subtracting a risk-seeking quantile. This originated from DSR (Petersen et al., 2021), where they encourage the model to factor in those well-fitted output expressions and ignore the poor-fitted output expressions. They proposed to subtract a $k\%$ of the quantile of the rewards instead of the empirical mean of the rewards.

$$\nabla_\theta J(\theta) \approx \frac{1}{N}\sum_{i=1}^{N}(R(\tau^i)-b)\nabla_\theta \log p_\theta(\tau^i), \qquad \text{where } b = k\% \text{ quantile of the rewards}$$

In Figure 2, b is set as 75% quantile of rewards, following the original hyper-parameter definition in DSR (Petersen et al., 2021).

D.3 Time Benchmark of DSR-Rex

We use three types of sequential decoders for the time benchmark setting. The major configurations are listed in Table 5.

General Parameters			
max length	20		
training dataset size	2048		
validation dataset size	2048		
total learning iterations	200		
Optimizer Hyperparameters			
optimizer	Adam		
learning rate	0.009		
entropy weight	0.03		
entropy gamma	0.7		
Decoder-relevant Hyperparameters			
choice of decoder	GRU	LSTM	Multi-head Self-Attention
num layers	3	3	3
hidden size	128	128	128
dropout	0.5	0.5	NA
number of head	NA	NA	6

Table 5: Hyperparameters for the RNN Model