SRM-LoRA: Sub-Riemannian-Style Updates for Mitigating LLM Hallucination in Low-Rank Adaptation

Anonymous Authors[1]

Abstract

Hallucination remains a central challenge for deploying large language models in factual and knowledge-grounded settings, where post-training must correct unreliable generations without degrading reliable behavior. We propose SRM-LoRA, a low-rank adaptation method that views hallucination mitigation as selective control over update directions in LoRA adapter space. SRM-LoRA introduces a soft sub-Riemannian-style restriction on admissible parameter-space updates, encouraging factual correction while discouraging unreliable changes. This provides a geometric perspective on hallucination mitigation while preserving the standard forward LoRA computation. As a result, SRM-LoRA improves factual reliability without additional inference cost.

1. Introduction

Large language models (LLMs) (Radford et al., 2019; Brown et al., 2020; Ouyang et al., 2022; Wei et al., 2023) have achieved strong performance across question answering, summarization, dialogue, and reasoning tasks, yet they still often generate fluent but unsupported or factually incorrect outputs. This phenomenon, commonly known as hallucination, is especially problematic in factual and knowledge-grounded settings where a model must rely on provided evidence. While supervised fine-tuning and preference optimization can reduce hallucination, broad post-training updates may also alter behaviors that are already reliable or reinforce distinctions that are not directly tied to factual correction (Gekhman et al., 2024; Sharma et al., 2025; Chowdhury et al., 2024)

In low-rank adaptation, different adapter directions can contribute differently to grounded and hallucinated continuations. Uniformly updating all LoRA directions may therefore be inefficient or even undesirable, especially when only a subset of directions is relevant to suppressing unreliable generations.

This motivates a geometric view of adapter optimization. In Riemannian optimization, the parameter space is equipped with a metric that determines how costly it is to move in each direction. A sub-Riemannian perspective further restricts or penalizes motion so that only selected directions are easily accessible, while movement along other directions becomes costly or indirect. For hallucination mitigation, this suggests that adapter updates should not freely move along all low-rank directions, but should instead favor grounded correction directions and penalize directions associated with unreliable continuations.

We propose **SRM-LoRA**, a low-rank adaptation method that implements this idea through a support-mask-based backward update rule. Given positive–negative pairs, SRM-LoRA identifies LoRA directions that are more strongly associated with hallucinated continuations and increases their effective update cost by suppressing their backward gradients. Importantly, the mask is not applied as a forward gate: the LoRA forward computation remains unchanged, so SRM-LoRA introduces no additional inference-time cost.

We evaluate SRM-LoRA on hallucination benchmarks and compare it against standard LoRA, contrastive LoRA, and forward-gated masking variants. These comparisons are designed to separate the effect of contrastive learning from the effect of direction-selective backward geometry.

Our contributions are as follows:

- We introduce SRM-LoRA, a sub-Riemannian-inspired LoRA framework for hallucination mitigation.
- We use positive–negative activation contrasts to identify grounded and hallucination-prone LoRA rank directions.
- We define a local metric from the directional scores and obtain sub-Riemannian-style backward gradient scaling through the inverse metric.

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the FoGen Workshop at ICML 2026. Do not distribute.

2. Related Work

2.1. LoRA Fine-Tuning

LoRA (Hu et al., 2021; Liu et al., 2024; Dettmers et al., 2023) has been widely adopted as an efficient parameter-efficient fine-tuning method. Prior works have shown that LoRA-based approaches can achieve strong performance while training only a small number of parameters and requiring substantially less training time than full fine-tuning (Hu et al., 2021; Ding et al., 2023; Whitehouse et al., 2024). Beyond its efficiency, LoRA offers a compact and structured adaptation space through its low-rank adapter matrices.

2.2. Riemannian Geometry for LoRA

Several recent works have explored Riemannian geometry in the context of LoRA. (Park et al., 2025) introduces Riemannian optimization to mitigate basis redundancy in LoRA, while (Bogachev et al., 2025) proposes an optimizer defined on a new manifold. In addition, (Zhang & Pilanci, 2024b) exploits the geometric benefits of Riemannian manifolds by multiplying small correction matrices, and (Li et al., 2025) similarly places part of the matrix structure on a manifold to leverage geometric information.

2.3. Limitations of Direct Sub-Riemannian Modeling

Prior sub-Riemannian methods have primarily been used from a control-theoretic perspective, where the geometry constrains motion to admissible directions over trajectories, curves, or spatially organized points (Jansson & Modin, 2025; Bellaard et al., 2023; Koo, 2023). However, applying such geometry directly to large-scale LLM post training is nontrivial: token representations and hidden states can be viewed as high-dimensional points, (Peters et al., 2018; Devlin et al., 2019) but computing geometric constraints over sample-, token-, and layer-level representations would introduce substantial memory and computational overhead.

3. Theory

3.1. Geometric Setup

Recent Riemannian approaches to LoRA view low-rank adapter parameters as geometrically structured optimization spaces. (Park et al., 2025; Zhang & Pilanci, 2024a) We use geometry for a different purpose: to define a sample-dependent local metric that assigns different update costs to different LoRA rank directions.

Let f_{θ_0} be a frozen pretrained language model. For a linear module ℓ, LoRA parameterizes the trainable update as

$$W_\ell = W_{0,\ell} + s_{\mathrm{LoRA}} B_\ell A_\ell, \tag{1}$$

where

$$A_\ell \in \mathbb{R}^{r\times d_{\mathrm{in}}}, \qquad B_\ell \in \mathbb{R}^{d_{\mathrm{out}}\times r}, \qquad r \ll \min(d_{\mathrm{in}}, d_{\mathrm{out}}).$$

For a hidden state $x_{\ell,b,t}$ at module ℓ, sample b, and token position t, define the local LoRA rank coordinate

$$z_{\ell,b,t} = A_\ell x_{\ell,b,t} \in \mathbb{R}^r. \tag{2}$$

The adapter response is

$$h^{\mathrm{LoRA}}_{\ell,b,t} = s_{\mathrm{LoRA}} B_\ell z_{\ell,b,t}. \tag{3}$$

We use $z_{\ell,b,t}$ as a local rank-coordinate representation through which adapter gradients flow. This coordinate view does not assume that the rank coordinates remain globally independent after multiplication by B_ℓ or through later transformer layers. It only gives a local tangent representation:

$$T_{z_{\ell,b,t}}\mathcal{M}_{\ell,b,t} \simeq \mathbb{R}^r, \tag{4}$$

with coordinate basis $\{e_k\}_{k=1}^r$.

The goal is to define a local metric on the tangent space in Eq. (4) such that grounded-dominant directions have low movement cost and hallucination-dominant directions have high movement cost. Through the chain rule, the resulting metric-gradient scaling on $z_{\ell,b,t}$ induces the corresponding backward scaling on the LoRA parameters.

3.2. Grounded Evidence from Good–Bad Contrast

For each training example, let y^+ denote a grounded continuation and y^- denote a hallucinated or counterfactual continuation under the same prompt. These two continuations induce local LoRA rank responses

$$z^+_{\ell,b,t,k} \qquad \text{and} \qquad z^-_{\ell,b,t,k}.$$

Definition 3.1 (Grounded activation contrast)**.** For each local coordinate (ℓ, b, t, k), define

$$s_{\ell,b,t,k} = \log \frac{|z^+_{\ell,b,t,k}| + \varepsilon}{|z^-_{\ell,b,t,k}| + \varepsilon}, \tag{5}$$

where $\varepsilon > 0$ is a small constant.

The sign of $s_{\ell,b,t,k}$ in Eq. (5) indicates whether coordinate k is more active under the grounded continuation or the hallucinated continuation.

Definition 3.2 (Soft grounded score)**.** Define

$$m_{\ell,b,t,k} = \sigma\left(\alpha(s_{\ell,b,t,k} - \tau)\right), \qquad m_{\ell,b,t,k} \in (0,1), \tag{6}$$

where $\alpha > 0$ controls sharpness and τ is a threshold.

The value $m_{\ell,b,t,k}$ in Eq. (6) is interpreted as a soft groundedness score for the local LoRA rank direction. Values above $1/2$ indicate grounded-dominant directions, values below $1/2$ indicate hallucination-dominant directions, and $1/2$ is neutral. The score estimates directional reliability before defining an update rule.

3.3. Soft Sub-Riemannian-Style Metric

A classical sub-Riemannian metric assigns finite cost to preferred directions and prohibitive cost to disallowed directions. SRM-LoRA uses a soft version of this idea by assigning coordinate-wise costs based on groundedness scores.

Definition 3.3 (Signed grounded evidence)**.** Define

$$q_{\ell,b,t,k} = 2m_{\ell,b,t,k} - 1. \tag{7}$$

Thus, $q > 0$ indicates grounded-dominant evidence, $q < 0$ indicates hallucination-dominant evidence, and $q = 0$ is neutral.

Definition 3.4 (Admissibility scale)**.** Let

$$a_{\ell,b,t,k} = \phi(q_{\ell,b,t,k}), \qquad a_{\ell,b,t,k} > 0, \tag{8}$$

where ϕ is a positive monotone increasing function satisfying

$$\phi(0) = 1, \qquad q_i > q_j \Rightarrow \phi(q_i) > \phi(q_j). \tag{9}$$

The value $a_{\ell,b,t,k}$ in Eq. (8) is a directional admissibility scale. Larger values indicate lower movement cost; smaller values indicate higher movement cost.

Definition 3.5 (Local anisotropic metric)**.** For tangent vector $u_{\ell,b,t} \in T_{z_{\ell,b,t}}\mathcal{M}_{\ell,b,t}$, define

$$g_{\ell,b,t}(u_{\ell,b,t}, u_{\ell,b,t}) = \sum_{k=1}^{r} \frac{u_{\ell,b,t,k}^2}{a_{\ell,b,t,k}}. \tag{10}$$

Equivalently, the local metric tensor is

$$G_{\ell,b,t} = \operatorname{diag}\left(\frac{1}{a_{\ell,b,t,1}}, \ldots, \frac{1}{a_{\ell,b,t,r}}\right). \tag{11}$$

The metric in Eq. (10) assigns low cost to grounded-dominant coordinates and high cost to hallucination-dominant coordinates.

Lemma 3.6 (Directional cost ordering)**.** *Let $a_{\ell,b,t,k} = \phi(q_{\ell,b,t,k})$ with ϕ positive and monotone increasing. If*

$$m_{\ell,b,t,i} > m_{\ell,b,t,j},$$

then the metric cost of a unit displacement along e_i is smaller than the metric cost of a unit displacement along e_j:

$$g_{\ell,b,t}(e_i, e_i) < g_{\ell,b,t}(e_j, e_j). \tag{12}$$

Proof. Since $m_i > m_j$, we have $q_i > q_j$. By monotonicity of ϕ, $a_i > a_j$. Since

$$g(e_k, e_k) = \frac{1}{a_k},$$

it follows that

$$g(e_i, e_i) = \frac{1}{a_i} < \frac{1}{a_j} = g(e_j, e_j).$$

□

Proposition 3.7 (Soft infinite-cost limit)**.** *Assume a sequence of hallucination-dominant coordinates has*

$$a_{\ell,b,t,k} \to 0.$$

Then for any tangent vector with nonzero component $u_{\ell,b,t,k} \neq 0$,

$$g_{\ell,b,t}(u_{\ell,b,t}, u_{\ell,b,t}) \to +\infty. \tag{13}$$

Proof. The metric in Eq. (10) contains the term

$$\frac{u_{\ell,b,t,k}^2}{a_{\ell,b,t,k}}.$$

If $u_{\ell,b,t,k} \neq 0$ and $a_{\ell,b,t,k} \to 0^+$, this term diverges to $+\infty$. Hence the full metric energy also diverges. □

Riemannian metric and sub-Riemannian-style limit. For finite $a_{\ell,b,t,k} > 0$, Eq. (10) defines a local anisotropic Riemannian metric. The sub-Riemannian-style restriction appears in the vanishing-admissibility limit of Proposition 3.7: finite-energy motion must have zero component along directions whose admissibility scale tends to zero. The implemented clipped scale keeps the metric finite for numerical stability; the geometric interpretation is a soft relaxation of the limiting restricted-motion geometry.

3.4. Riemannian Gradient under the Local Metric

We now derive why the mask-derived scale appears in backpropagation. A metric changes the steepest descent direction without changing the forward coordinate.

Let $\mathcal{L}$ be a differentiable scalar objective on the local coordinate space. The Euclidean gradient $\nabla^E_{z_{\ell,b,t}}\mathcal{L}$ is defined with respect to the standard inner product. Under the local metric $g_{\ell,b,t}$, the Riemannian gradient is the unique tangent vector satisfying

$$g_{\ell,b,t}\left(\nabla^g_{z_{\ell,b,t}}\mathcal{L}, v\right) = d\mathcal{L}[v] \qquad \forall v \in T_{z_{\ell,b,t}}\mathcal{M}_{\ell,b,t}. \tag{14}$$

Lemma 3.8 (Inverse-metric gradient)**.** *Under the local metric tensor $G_{\ell,b,t}$ in Eq. (11), the Riemannian gradient is*

$$\nabla^g_{z_{\ell,b,t}}\mathcal{L} = G^{-1}_{\ell,b,t}\nabla^E_{z_{\ell,b,t}}\mathcal{L}. \tag{15}$$

Equivalently, for each coordinate k,

$$\left(\nabla^g_{z_{\ell,b,t}}\mathcal{L}\right)_k = a_{\ell,b,t,k}\left(\nabla^E_{z_{\ell,b,t}}\mathcal{L}\right)_k. \tag{16}$$

Proof. In local coordinates,

$$d\mathcal{L}[v] = \left\langle \nabla^E_{z_{\ell,b,t}}\mathcal{L}, v \right\rangle.$$

Also,

$$g_{\ell,b,t}(u,v) = u^\top G_{\ell,b,t} v.$$

Substituting $u = \nabla^g \mathcal{L}$ into Eq. (14) yields

$$(\nabla^g \mathcal{L})^\top G_{\ell,b,t} v = (\nabla^E \mathcal{L})^\top v \qquad \forall v.$$

Therefore,

$$G_{\ell,b,t} \nabla^g \mathcal{L} = \nabla^E \mathcal{L},$$

and hence

$$\nabla^g \mathcal{L} = G^{-1}_{\ell,b,t} \nabla^E \mathcal{L}.$$

Because

$$G^{-1}_{\ell,b,t} = \operatorname{diag}\left(a_{\ell,b,t,1}, \ldots, a_{\ell,b,t,r}\right),$$

the coordinate-wise expression follows. □

Lemma 3.8 gives the mathematical connection to backpropagation. Once the mask-derived score defines the metric tensor $G_{\ell,b,t}$ in Eq. (11), the induced gradient is the inverse-metric gradient in Eq. (15). Therefore, the mask appears as a multiplicative factor on the backward gradient.

3.5. Main Result

Theorem 3.9 (Mask-induced inverse-metric backpropagation)**.** *Let $m_{\ell,b,t,k}$ be the soft grounded score in Eq. (6). Let $a_{\ell,b,t,k} = \phi(2m_{\ell,b,t,k} - 1)$ be a positive monotone admissibility scale, and let $g_{\ell,b,t}$ be the local anisotropic metric in Eq. (10). Then steepest descent under $g_{\ell,b,t}$ follows the direction*

$$-\nabla^g_{z_{\ell,b,t}} \mathcal{L} = -G^{-1}_{\ell,b,t} \nabla^E_{z_{\ell,b,t}} \mathcal{L}. \tag{17}$$

In coordinates,

$$-\left(\nabla^g_{z_{\ell,b,t}} \mathcal{L}\right)_k = -a_{\ell,b,t,k} \frac{\partial \mathcal{L}}{\partial z_{\ell,b,t,k}}. \tag{18}$$

Proof. Steepest descent under a Riemannian metric follows the negative Riemannian gradient. By Lemma 3.8,

$$-\nabla^g \mathcal{L} = -G^{-1} \nabla^E \mathcal{L}.$$

The coordinate-wise expression in Eq. (18) follows from the diagonal form of G^{-1}. □

3.6. A Concrete Scale Family

The previous results hold for any positive monotone scale $a = \phi(q)$. A convenient scale family is

$$a_{\ell,b,t,k} = \operatorname{clip}\left(\left[1 + \gamma_{\mathrm{SR}}(2m_{\ell,b,t,k} - 1)\right]^\beta, a_{\min}, a_{\max}\right), \tag{19}$$

with $\gamma_{\mathrm{SR}} \geq 0$, $\beta \geq 0$, and $0 < a_{\min} \leq a_{\max}$. The scale in Eq. (19) preserves the neutral point

$$m_{\ell,b,t,k} = \frac{1}{2} \quad \Rightarrow \quad a_{\ell,b,t,k} = 1,$$

amplifies grounded-dominant directions, and suppresses hallucination-dominant directions.

Substituting Eq. (19) into the coordinate inverse-metric gradient in Eq. (16) gives

$$\frac{\partial \mathcal{L}}{\partial z_{\ell,b,t,k}} \mapsto a_{\ell,b,t,k} \frac{\partial \mathcal{L}}{\partial z_{\ell,b,t,k}}. \tag{20}$$

Eq. (20) is the coordinate form of Riemannian steepest descent under the proposed local metric. The mask parameterizes the metric tensor, whose inverse scales gradients during backpropagation.

4. Method

4.1. Overview

SRM-LoRA adapts a frozen language model using LoRA and modifies the backward geometry of LoRA rank coordinates. Each training example consists of a prompt x, a factual continuation y^+, and a hallucinated or counterfactual continuation y^-. The objective is to increase the model preference for y^+ over y^- while suppressing LoRA rank directions that are more strongly associated with the hallucinated continuation.

SRM-LoRA proceeds in three steps. First, it runs the positive and negative continuations through the LoRA model and compares their local LoRA rank activations. Second, it converts the positive–negative activation contrast into a detached metric scale. Third, it applies the metric scale to the backward gradients flowing through the LoRA rank coordinates. The mask is not used as a forward activation gate.

4.2. Length-Normalized Margin Objective

For a continuation $y = (y_1, \ldots, y_T)$ under prompt x, we use the length-normalized log-likelihood

$$\bar{\ell}_\theta(y \mid x) = \frac{1}{T} \sum_{t=1}^{T} \log p_\theta(y_t \mid x, y_{<t}). \tag{21}$$

This avoids making the margin depend directly on continuation length. The positive–negative preference gap is

$$\Delta_\theta(x, y^+, y^-) = \bar{\ell}_\theta(y^+ \mid x) - \bar{\ell}_\theta(y^- \mid x). \tag{22}$$

SRM-LoRA uses the margin loss

$$\mathcal{L}_{\mathrm{SRM}} = \lambda \left[\gamma - \Delta_\theta(x, y^+, y^-)\right]_+, \tag{23}$$

or equivalently,

$$\mathcal{L}_{\mathrm{SRM}} = \lambda \left[\gamma + \bar{\ell}_\theta(y^- \mid x) - \bar{\ell}_\theta(y^+ \mid x)\right]_+. \tag{24}$$

The loss is zero when the positive continuation is preferred by at least margin γ.

4.3. LoRA Rank Activations

For a LoRA-augmented module ℓ, the LoRA branch computes

$$z_{\ell,b,t} = A_\ell x_{\ell,b,t} \in \mathbb{R}^r, \tag{25}$$

where b is the sample index and t is the token position. The low-rank response is

$$h^{\mathrm{LoRA}}_{\ell,b,t} = s_{\mathrm{LoRA}} B_\ell z_{\ell,b,t}. \tag{26}$$

For each training pair, SRM-LoRA obtains two rank-activation tensors from the same two forward passes used to compute the margin loss:

$$z^+_{\ell,b,t,k} \qquad \text{and} \qquad z^-_{\ell,b,t,k}.$$

Here, z^+ is induced by the factual continuation and z^- is induced by the hallucinated continuation.

4.4. Token Alignment and Stop-Gradient Mask Construction

Positive and negative continuations can have different lengths. For sample b, let s_b be the first continuation-token index, and let T_b^+ and T_b^- be the real unpadded sequence lengths for the positive and negative branches. SRM-LoRA computes the activation contrast only on the common continuation span:

$$t \in [s_b, \min(T_b^+, T_b^-)). \tag{27}$$

Prompt tokens, padding tokens, and unmatched suffix tokens receive the neutral mask value $m = 0.5$, which gives unit metric scale under signed-positive scaling. This prevents length mismatch from creating accidental amplification or suppression.

The mask is constructed with stop-gradient activations:

$$\tilde{z}^+ = \mathrm{sg}(z^+), \qquad \tilde{z}^- = \mathrm{sg}(z^-). \tag{28}$$

The mask and the resulting metric scale are treated as data-dependent coefficients during the current update. No gradient is propagated through the mask construction path.

For each aligned coordinate, SRM-LoRA computes

$$s_{\ell,b,t,k} = \log \frac{|\tilde{z}^+_{\ell,b,t,k}| + \varepsilon}{|\tilde{z}^-_{\ell,b,t,k}| + \varepsilon}, \tag{29}$$

and maps the score to a soft mask:

$$m_{\ell,b,t,k} = \sigma\left(\alpha(s_{\ell,b,t,k} - \tau)\right). \tag{30}$$

Values above 0.5 indicate grounded-dominant rank directions, values below 0.5 indicate hallucination-prone rank directions, and 0.5 is neutral.

Algorithm 1 SRM-LoRA Training Step

1. **Input:** prompt x, factual continuation y^+, negative continuation y^-; frozen base model; LoRA parameters $\{A_\ell, B_\ell\}$.
2. **Hyperparameters:** margin γ, loss weight λ, mask parameters α, τ, and metric parameters $\gamma_{\mathrm{SR}}, \beta, a_{\min}, a_{\max}$.
3. Forward (x, y^+) and (x, y^-) to obtain log-likelihoods and rank activations z^+ and z^-.
4. Compute the margin loss $\mathcal{L}_{\mathrm{SRM}}$ using Eq. (24).
5. Construct stop-gradient activations $\tilde{z}^+ = \mathrm{sg}(z^+)$ and $\tilde{z}^- = \mathrm{sg}(z^-)$.
6. For each aligned coordinate (ℓ, b, t, k), compute $s_{\ell,b,t,k}$ by Eq. (29), $m_{\ell,b,t,k}$ by Eq. (30), and $q_{\ell,b,t,k} = 2m_{\ell,b,t,k} - 1$.
7. Compute $a^+_{\ell,b,t,k}$ and $a^-_{\ell,b,t,k}$ using Eqs. (32)–(33).
8. Backpropagate $\mathcal{L}_{\mathrm{SRM}}$ with Eqs. (34)–(35).
9. Update only the LoRA parameters $\{A_\ell, B_\ell\}$.

4.5. Branch-Specific Metric Scaling

The mask is converted into signed evidence:

$$q_{\ell,b,t,k} = 2m_{\ell,b,t,k} - 1. \tag{31}$$

For the positive branch, SRM-LoRA uses

$$a^+_{\ell,b,t,k} = \mathrm{clip}\left(\left[1 + \gamma_{\mathrm{SR}} q_{\ell,b,t,k}\right]^\beta, a_{\min}, a_{\max}\right). \tag{32}$$

For the negative branch, SRM-LoRA uses the complementary scale:

$$a^-_{\ell,b,t,k} = \mathrm{clip}\left(\left[1 - \gamma_{\mathrm{SR}} q_{\ell,b,t,k}\right]^\beta, a_{\min}, a_{\max}\right). \tag{33}$$

Thus, grounded-dominant directions are amplified on the positive branch, while negative-dominant directions can be emphasized on the negative branch under the margin objective.

During backpropagation, SRM-LoRA applies

$$\frac{\partial \mathcal{L}_{\mathrm{SRM}}}{\partial z^+_{\ell,b,t,k}} \leftarrow a^+_{\ell,b,t,k} \frac{\partial \mathcal{L}_{\mathrm{SRM}}}{\partial z^+_{\ell,b,t,k}}, \tag{34}$$

and

$$\frac{\partial \mathcal{L}_{\mathrm{SRM}}}{\partial z^-_{\ell,b,t,k}} \leftarrow a^-_{\ell,b,t,k} \frac{\partial \mathcal{L}_{\mathrm{SRM}}}{\partial z^-_{\ell,b,t,k}}. \tag{35}$$

This implements the inverse-metric gradient derived in Section 3. The forward LoRA activation itself is not multiplied by the mask.

4.6. Training Step

One SRM-LoRA update is summarized below.

The two forward passes provide both the losses and the rank activations; no additional forward pass is needed to recompute the log-likelihoods.

4.7. Cost and Relation to Forward SoftMask

SRM-LoRA requires one positive and one negative forward pass per update, so its forward cost is approximately twice that of a single gold-answer LoRA update. The same two passes provide the activations used to build the metric scale.

A forward SoftMask baseline can use the same mask but applies it to the LoRA activation:

$$z_{\ell,b,t,k} \leftarrow m_{\ell,b,t,k} z_{\ell,b,t,k}. \tag{36}$$

SRM-LoRA instead applies the mask-derived scale to the backward gradient:

$$\nabla_z \mathcal{L}_{\mathrm{SRM}} \leftarrow a \odot \nabla_z \mathcal{L}_{\mathrm{SRM}}. \tag{37}$$

Thus, forward SoftMask tests the mask as a representation gate, while SRM-LoRA uses the mask as a local metric for adaptation.

5. Experiments

We evaluate SRM-LoRA on hallucination-sensitive and knowledge-grounded generation tasks. The experiments are designed to answer three questions: (i) whether SRM-LoRA reduces hallucination relative to a frozen base model, (ii) whether the improvement is stronger than standard LoRA fine-tuning, and (iii) whether the proposed backward metric update differs empirically from a forward activation mask or a contrastive-only update using the same positive–negative supervision.

5.1. Experimental Setup

Base model. All experiments use Qwen2.5-7B-Instruct as the base language model. The base model is frozen except for the LoRA adapter parameters. Unless otherwise specified, LoRA is inserted into attention and MLP projection modules, and only the selected LoRA parameters are updated.

Training data. Training uses HaluEval QA examples. For each example, the gold answer is used as the positive continuation. For contrastive methods, the negative continuation is generated on the fly by the frozen base model and accepted only when the frozen judge labels it as hallucinated. This makes the negative branch model-derived rather than fixed to a dataset-provided hallucinated answer.

Evaluation datasets. We evaluate the main checkpoint trajectories on HaluEval dialogue, HaluEval summarization, and DROP. We additionally include a step-150 comparison against Contrastive LoRA on HotpotQA-fullwiki to test robustness beyond the HaluEval training distribution.

Baselines. We compare SRM-LoRA against three baselines.

- **Plain LoRA**: standard LoRA fine-tuning on the gold answer using cross-entropy only. This baseline tests whether ordinary parameter-efficient supervised adaptation already reduces hallucination.
- **Forward SoftMask**: a mask-based LoRA baseline that uses the same positive–negative activation contrast as SRM-LoRA, but applies the resulting mask as a forward gate on the LoRA rank activation. This baseline tests whether the mask signal helps merely as a representation filter.
- **Contrastive LoRA**: a contrastive-only baseline that uses the same positive and negative continuations as SRM-LoRA, but does not use a mask or metric-induced gradient scaling. This baseline isolates whether the gains come only from the pairwise contrastive objective.

SRM-LoRA uses the same positive–negative contrast signal as the mask-based and contrastive baselines, but applies the induced scale to the backward gradient rather than to the forward activation. Therefore, the comparison with Forward SoftMask isolates representation gating versus backward metric scaling, while the comparison with Contrastive LoRA isolates metric-induced updates versus a contrastive-only objective.

5.2. Evaluation Metrics

For each evaluation example, we compare the adapted model against the frozen base model using a semantic hallucination judge. The judge returns either `HALLUCINATED` or `NOT_HALLUCINATED`. We report three quantities.

Hallucination rate. The hallucination rate is

$$\mathrm{HallRate} = \frac{\#\mathrm{Hallucinated}}{\#\mathrm{Evaluated}}.$$

Lower hallucination rate indicates better factual reliability.

Improved and worsened examples. An example is counted as improved if the base model is judged hallucinated but the adapted model is judged not hallucinated:

$$\mathrm{Improved} = \#\{\, i : \mathrm{Base}_i = \mathrm{Hallucinated}, \; \mathrm{Adapted}_i = \mathrm{NotHallucinated} \,\}.$$

An example is counted as worsened if the base model is judged not hallucinated but the adapted model is judged hallucinated:

$$\mathrm{Worsened} = \#\{\, i : \mathrm{Base}_i = \mathrm{NotHallucinated}, \; \mathrm{Adapted}_i = \mathrm{Hallucinated} \,\}.$$

Table 1. Main results across training checkpoints. Each checkpoint cell reports Net followed by the hallucination rate in parentheses. Net is the number of examples corrected by the adapted model minus the number of examples corrupted relative to the base model. The hallucination rate is computed as $\#\text{Hallucinated}/\#\text{Evaluated}$, where the numerator is the number of evaluated examples judged hallucinated and the denominator is the number of used evaluation examples. For the Average row, Net is averaged over datasets and the hallucination rate is computed over all evaluated examples across the three datasets.

Dataset	Method	Metric	Step 50	Step 100	Step 150
HaluEval (dialogue) (Li et al., 2023)	Plain LoRA	Net (Hall. rate)	15 (0.7980)	43 (0.7952)	50 (0.7945)
	Forward SoftMask	Net (Hall. rate)	34 (0.7963)	58 (0.7939)	**103 (0.7894)**
	SRM-LoRA (Ours)	Net (Hall. rate)	23 (0.7972)	**60 (0.7935)**	86 (0.7909)
HaluEval (summ.) (Li et al., 2023)	Plain LoRA	Net (Hall. rate)	-5 (0.4314)	2 (0.4307)	-19 (0.4328)
	Forward SoftMask	Net (Hall. rate)	15 (0.4293)	24 (0.4284)	34 (0.4274)
	SRM-LoRA (Ours)	Net (Hall. rate)	**29 (0.4278)**	**31 (0.4276)**	**63 (0.4244)**
DROP (Dua et al., 2019)	Plain LoRA	Net (Hall. rate)	5 (0.3993)	67 (0.3928)	81 (0.3913)
	Forward SoftMask	Net (Hall. rate)	**41 (0.3956)**	**97 (0.3897)**	102 (0.3892)
	SRM-LoRA (Ours)	Net (Hall. rate)	39 (0.3961)	84 (0.3914)	**119 (0.3877)**
Average	Plain LoRA	Avg. Net (pooled Hall. rate)	5.00 (0.5451)	37.33 (0.5419)	37.33 (0.5419)
	Forward SoftMask	Avg. Net (pooled Hall. rate)	30.00 (0.5427)	**59.67 (0.5397)**	79.67 (0.5376)
	SRM-LoRA (Ours)	Avg. Net (pooled Hall. rate)	**30.33 (0.5426)**	58.33 (0.5398)	**89.33 (0.5367)**

Net improvement. The main correction metric is

$$\text{Net} = \text{Improved} - \text{Worsened}.$$

Net improvement is important because a model can improve many hallucinated examples while also corrupting many originally correct examples. A higher Net indicates that the adaptation corrects more errors than it introduces.

5.3. Main Results

HaluEval dialogue. On HaluEval dialogue, Forward SoftMask achieves the best final Net at step 150, with 103 corrected net examples and a hallucination rate of 0.7894. SRM-LoRA performs best at step 100, with Net 60 and hallucination rate 0.7935, but falls behind Forward SoftMask at step 150. This shows that the forward gate can be strong on dialogue-style outputs, where suppressing certain rank responses may directly reduce unreliable generations.

HaluEval summarization. On HaluEval summarization, SRM-LoRA consistently outperforms both baselines across all checkpoints. At step 150, SRM-LoRA reaches Net 63 and a hallucination rate of 0.4244, while Forward SoftMask reaches Net 34 and Plain LoRA degrades to Net -19. This result is important because summarization often requires preserving reference faithfulness while avoiding unsupported additions. The improvement suggests that SRM-LoRA's backward metric update can correct hallucination without over-constraining the forward representation.

DROP. On DROP, Forward SoftMask performs best at steps 50 and 100, but SRM-LoRA achieves the best final result at step 150. SRM-LoRA obtains Net 119 and a hallucination rate of 0.3877, compared to Net 102 and hallucination rate 0.3892 for Forward SoftMask. This indicates that SRM-LoRA can be especially effective after longer adaptation on reference-grounded question answering.

5.4. Improved, Worsened, and Net Counts

Figure 1 decomposes Net improvement into improved and worsened counts. This decomposition is useful because Net alone does not show whether a method improves by correcting many hallucinated examples, by avoiding corruption of correct examples, or both.

Final-step behavior. At step 150, SRM-LoRA has the highest Net on HaluEval summarization and DROP, while Forward SoftMask has the highest Net on HaluEval dialogue. This pattern suggests that the two mask-based methods behave differently: forward masking can be effective when suppressing unreliable activations directly helps generation, while SRM-LoRA is stronger when the update geometry must preserve the forward representation while changing how the adapter learns.

5.5. Step-150 Comparison with Contrastive LoRA

Table 2 compares SRM-LoRA with Contrastive LoRA at step 150. This comparison directly tests whether the positive–negative contrastive objective is sufficient, or whether the proposed sub-Riemannian-style metric update provides additional robustness.

Near-training versus out-of-distribution behavior. The step-150 comparison shows a clear difference between near-training and out-of-distribution evaluations. On HaluEval dialogue and HaluEval summarization, which are closer to the HaluEval QA training distribution, Contrastive LoRA is highly competitive and slightly stronger in Net improvement: it obtains Net 100 versus 98 on HaluEval dialogue and Net 44 versus 35 on HaluEval summarization. This suggests that when the evaluation format is close to the training distri-

Figure 1. Improved, worsened, and net improved example counts across training steps.

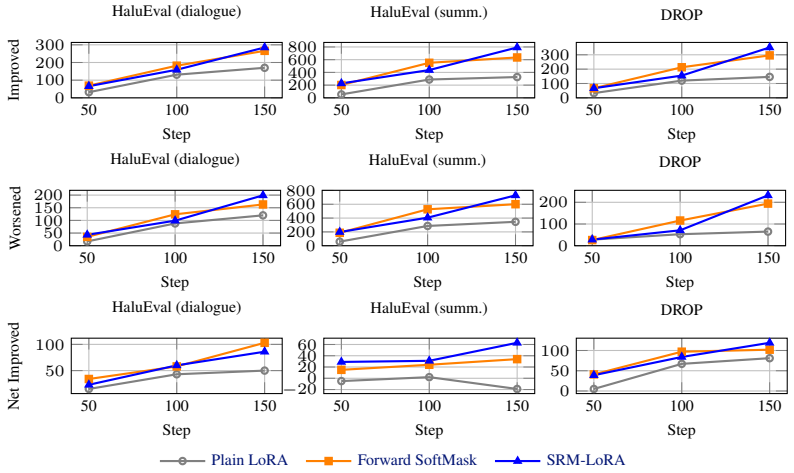

Table 2. Step-150 comparison between SRM-LoRA and Contrastive LoRA on hallucination-sensitive evaluations.

Dataset	Method	Eval Dataset Size	Hallucination Rate	Improved	Worsened	Net
HaluEval Dialogue	SRM-LoRA (Ours)	10000	0.7910	270	172	98
	Contrastive LoRA	10000	0.7908	262	162	100
HaluEval Summarization	SRM-LoRA (Ours)	10000	0.4272	630	595	35
	Contrastive LoRA	10000	0.4273	593	549	44
DROP	SRM-LoRA (Ours)	9535	0.3871	304	180	124
	Contrastive LoRA	9535	0.3883	293	179	114
HotpotQA-fullwiki	SRM-LoRA (Ours)	7395	0.4748	336	151	185
	Contrastive LoRA	7395	0.4787	311	155	156

bution, directly optimizing the positive–negative contrastive objective can be sufficient.

In contrast, SRM-LoRA is stronger on the more out-of-distribution reference-grounded reasoning datasets. On DROP, SRM-LoRA obtains Net 124 compared to 114 for Contrastive LoRA, and it also achieves a lower hallucination rate, 0.3871 versus 0.3883. On HotpotQA-fullwiki, the gap is larger: SRM-LoRA reaches Net 185 compared to 156 for Contrastive LoRA, while also reducing hallucination rate from 0.4787 to 0.4748. These results suggest that the sub-Riemannian-style update is not merely reproducing a contrastive objective. Instead, the metric-induced scaling appears to provide a more robust update rule when the evaluation distribution differs from the HaluEval QA training data.

6. Conclusion

We presented SRM-LoRA, a sub-Riemannian-inspired LoRA framework for hallucination mitigation. SRM-LoRA uses positive–negative activation contrasts to identify grounded-dominant and hallucination-prone LoRA rank directions. Instead of applying this signal as a forward gate, the method interprets it as a local metric over adapter coordinates and applies the inverse metric through backward gradient scaling.

This view separates the supervision signal from the update geometry: positive–negative pairs define directional evidence, while the local metric determines how gradients move through LoRA space. SRM-LoRA therefore reframes hallucination mitigation from objective-level optimization to geometry-aware adapter adaptation.

References

Bellaard, G., Bon, D. L. J., Pai, G., Smets, B. M. N., and Duits, R. Analysis of (sub-)riemannian pde-g-cnns. *Journal of Mathematical Imaging and Vision*, 65:819–843, 2023. doi: 10.1007/s10851-023-01147-w. URL https://doi.org/10.1007/s10851-023-01147-w.

Bogachev, V., Aletov, V., Molozhavenko, A., Bobkov, D., Soboleva, V., Alanov, A., and Rakhuba, M. Lora meets riemannion: Muon optimizer for parametrization-independent low-rank adapters, 2025.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G.,

Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language models are few-shot learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Chowdhury, S. R., Kini, A., and Natarajan, N. Provably robust dpo: Aligning language models with noisy feedback, 2024. URL https://arxiv.org/abs/2403.00409.

Dettmers, T., Pagnoni, A., Holtzman, A., and Zettlemoyer, L. Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*, 2023.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186. Association for Computational Linguistics, 2019.

Ding, N., Lv, X., Wang, Q., Chen, Y., Zhou, B., Liu, Z., and Sun, M. Sparse low-rank adaptation of pre-trained language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 4133–4145, December 2023.

Dua, D., Wang, Y., Dasigi, P., Stanovsky, G., Singh, S., and Gardner, M. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In Burstein, J., Doran, C., and Solorio, T. (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 2368–2378, 2019.

Gekhman, Z., Yona, G., Aharoni, R., Eyal, M., Feder, A., Reichart, R., and Herzig, J. Does fine-tuning LLMs on new knowledge encourage hallucinations? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 7765–7784, Miami, Florida, USA, 2024. Association for Computational Linguistics.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. Lora: Low-rank adaptation of large language models, 2021. URL https://arxiv.org/abs/2106.09685.

Jansson, E. and Modin, K. Sub-riemannian landmark matching and its interpretation as residual neural networks. *Journal of Computational Dynamics*, 12(3):467–490, 2025. ISSN 2158-2505. doi: 10.3934/jcd.2025004.

Koo, J. Sinogram upsampling via sub-riemannian diffusion with adaptive weighting. *Electronics*, 12(21):4503, 2023. doi: 10.3390/electronics12214503. URL https://doi.org/10.3390/electronics12214503.

Li, J., Cheng, X., Zhao, X., Nie, J.-Y., and Wen, J.-R. HaluEval: A large-scale hallucination evaluation benchmark for large language models. In Bouamor, H., Pino, J., and Bali, K. (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 6449–6464, 2023.

Li, Z., Sajadmanesh, S., Li, J., and Lyu, L. Stella: Subspace learning in low-rank adaptation using stiefel manifold. In *Advances in Neural Information Processing Systems*, volume 38. Curran Associates, Inc., 2025.

Liu, S.-Y., Wang, C.-Y., Yin, H., Molchanov, P., Wang, Y.-C. F., Cheng, K.-T., and Chen, M.-H. Dora: Weight-decomposed low-rank adaptation. *arXiv preprint arXiv:2402.09353*, 2024.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P. F., Leike, J., and Lowe, R. Training language models to follow instructions with human feedback. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 27730–27744. Curran Associates, Inc., 2022.

Park, J., Kang, M., Lee, S., Lee, H., Kim, S., and Lee, J. Riemannian optimization for LoRA on the stiefel manifold. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 20971–20985, 2025.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 2227–2237. Association for Computational Linguistics, June 2018.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. Language models are unsupervised multitask learners. 2019.

Sharma, M., Tong, M., Korbak, T., Duvenaud, D., Askell, A., Bowman, S. R., Cheng, N., Durmus, E., Hatfield-Dodds, Z., Johnston, S. R., Kravec, S., Maxwell, T., McCandlish, S., Ndousse, K., Rausch, O., Schiefer, N., Yan, D., Zhang, M., and Perez, E. Towards understanding sycophancy in language models, 2025. URL https://arxiv.org/abs/2310.13548.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q., and Zhou, D. Chain-of-thought prompting elicits reasoning in large language models, 2023. URL https://arxiv.org/abs/2201.11903.

Whitehouse, C., Huot, F., Bastings, J., Dehghani, M., Lin, C.-C., and Lapata, M. Low-rank adaptation for multilingual summarization: An empirical study. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 1202–1228, June 2024.

Zhang, F. and Pilanci, M. Riemannian preconditioned lora for fine-tuning foundation models, 2024a.

Zhang, F. and Pilanci, M. Riemannian preconditioned lora for fine-tuning foundation models, 2024b.

A. Appendix Overview

This appendix provides implementation and reproducibility details for SRM-LoRA. We include the exact prompt templates, the evaluation judge prompt, dataset normalization rules, negative-continuation construction, training objectives, hyperparameters, ablations, diagnostic traces, and computational cost. The goal is to make the distinction between the objective, the mask construction, and the metric-induced backward update explicit.

SRM-LoRA uses positive–negative continuation pairs to estimate which LoRA rank directions are grounded-dominant or hallucination-dominant. Unlike a forward masking method, SRM-LoRA does not multiply the LoRA activation by the mask during inference. Instead, it converts the activation contrast into a local metric and applies the induced inverse-metric scale to the backward gradient during training.

B. Prompt Templates

All training, negative sampling, and evaluation calls use the same answer prompt format. This avoids a train–evaluation prompt mismatch. When the tokenizer provides a chat template, the system and user messages below are rendered through the tokenizer's chat template. Otherwise, they are rendered as a plain text prompt with explicit system, user, and assistant fields.

B.1. Reference-Grounded Answer Prompt

For examples with a reference field, the model is instructed to answer only from the provided reference and to return only the final answer.

Reference-Grounded Answer Prompt

System:
You are a precise question-answering assistant. Answer the user's question using only the provided reference. Do not explain your reasoning. Do not restate the question. Return only the final answer text. If the answer is a number, return only the number.

User:
REFERENCE:
`{reference}`
QUESTION:
`{question}`
Return only the final answer.

Assistant:

B.2. Answer Prompt without Reference

For examples without a reference field, the reference block is omitted while the concise final-answer instruction is preserved.

Open Answer Prompt

System:
You are a precise question-answering assistant. Answer the user's question as accurately as possible. Do not explain your reasoning. Do not restate the question. Return only the final answer text. If the answer is a number, return only the number.

User:
QUESTION:
`{question}`
Return only the final answer.

Assistant:

B.3. Hallucination Judge Prompt

Evaluation uses an LLM-based semantic judge. The judge receives the question, gold answer, optional reference, and model prediction. It must return exactly one label: `NOT_HALLUCINATED` or `HALLUCINATED`. Harmless paraphrases are allowed. Predictions are marked hallucinated when they contradict the gold answer, fail to answer the question, or add incorrect factual claims.

Table 3. Training and evaluation dataset normalization.

Dataset	Usage	Split	Input / question field	Reference field / answer field
HaluEval QA	Train	data	`question`	`knowledge` / `right_answer`
HaluEval dialogue	Eval	data	`dialogue_history`	`knowledge` / `right_response`
HaluEval summarization	Eval	data	Fixed summarization instruction	`document` / `right_summary`
DROP	Eval	validation	`question`	`passage` / `answers_spans.spans`
HotpotQA fullwiki	Eval	validation	`question`	`context` / `answer`

Hallucination Judge Prompt

You are a strict hallucination judge.
Given QUESTION, GOLD_ANSWER, optional REFERENCE, and MODEL_PREDICTION, decide whether MODEL_PREDICTION is hallucinated.
Return NOT_HALLUCINATED if the prediction gives the same final answer as GOLD_ANSWER, allowing harmless paraphrases.
Return HALLUCINATED if the prediction contradicts GOLD_ANSWER, fails to answer the question, or adds incorrect factual claims.
Output only one label: NOT_HALLUCINATED or HALLUCINATED.

QUESTION:
{`question`}

GOLD_ANSWER:
{`gold_answer`}

REFERENCE:
{`reference`}

MODEL_PREDICTION:
{`prediction`}

LABEL:

When chat formatting is available, the judge prompt is wrapped with the following system message.

Judge Chat Wrapper System Message

You are a strict hallucination judge. Output only the requested label.

The parser accepts a judge output only if a unique label can be extracted. If the judge output cannot be parsed, the sample is treated as a hallucinated failure in the evaluation denominator rather than being silently removed.

C. Datasets and Normalization

We train on HaluEval QA and evaluate on four hallucination-sensitive and knowledge-grounded benchmarks: HaluEval dialogue, HaluEval summarization, DROP, and HotpotQA fullwiki. Each dataset is normalized into a common example format:

$$(\texttt{question}, \texttt{reference}, \texttt{answer}, \texttt{hallucinated\_answer}, \texttt{acceptable\_answers}).$$

Rows with missing canonical fields are dropped. We do not use defensive fallback field chains: each dataset branch reads only the canonical columns listed in Table 3.

For HaluEval dialogue, the dialogue history is treated as the question-like input, the knowledge field is used as the reference, and the right response is the gold answer. For HaluEval summarization, the question field is replaced by a fixed task instruction: "Summarize the document using only the given reference." The document is used as the reference and the right summary is used as the gold answer. For DROP, the passage is used as the reference and all answer spans are stored as acceptable answers. For HotpotQA fullwiki, the multi-document context is converted into a reference string by concatenating document titles and sentence lists.

Training data. Training uses HaluEval QA. The canonical training fields are `question`, `knowledge`, and `right_answer`. Although the HaluEval QA dataset may include a dataset-provided hallucinated answer, the contrastive, SoftMask, and SRM-LoRA training paths construct the negative continuation on the fly by sampling from the

Table 4. Negative-continuation sampling settings.

Parameter	Value
Negative sampler	Frozen base model with LoRA disabled
Prompt	Same answer-generation prompt as training/evaluation
Maximum sampling attempts	4
Sampling temperature	0.8
Top-p	0.95
Acceptance criterion	Frozen judge returns `HALLUCINATED`
Fallback	Skip contrastive update if no accepted negative is found

frozen base model. Plain LoRA does not use a hallucinated continuation.

D. Negative Continuation Construction

For methods that require positive–negative pairs, the positive continuation y^+ is the gold answer and the negative continuation y^- is sampled from the frozen base model. During negative sampling, LoRA adapters are disabled so that the negative side is produced by the original base model rather than by the currently adapted model.

For each training example, we sample up to A candidate continuations using the same answer-generation prompt used in training and evaluation. Each candidate is judged by the frozen base-model judge. We keep the first candidate judged as `HALLUCINATED`. If no hallucinated candidate is found within the sampling budget, the example is skipped for that contrastive update.

This procedure keeps the construction self-contained: the method does not rely on an external hallucination oracle beyond the dataset gold answer and the frozen model's own judge. It also prevents the adapted model from changing the negative sampler or the judge during training.

E. Training Objectives and Baselines

E.1. Plain LoRA

Plain LoRA trains only on the gold continuation with standard answer cross-entropy. Prompt tokens are masked from the loss, so the objective is computed only over continuation tokens. The gold answer is terminated with the tokenizer EOS token so that the model learns both the answer and the stopping condition.

E.2. Contrastive LoRA

Contrastive LoRA uses the same positive–negative pairs as SRM-LoRA but does not use a mask or a metric. For a continuation $y = (y_1, \dots, y_T)$ under prompt x, we define the length-normalized log-likelihood

$$\bar{\ell}_\theta(y \mid x) = \frac{1}{T} \sum_{t=1}^{T} \log p_\theta(y_t \mid x, y_{<t}).$$

The positive–negative gap is

$$\Delta_\theta(x, y^+, y^-) = \bar{\ell}_\theta(y^+ \mid x) - \bar{\ell}_\theta(y^- \mid x).$$

The contrastive margin loss is

$$\mathcal{L}_{\text{margin}} = \lambda \left[\gamma - \Delta_\theta(x, y^+, y^-)\right]_+ = \lambda \left[\gamma + \bar{\ell}_\theta(y^- \mid x) - \bar{\ell}_\theta(y^+ \mid x)\right]_+ .$$

This baseline tests whether the positive–negative objective alone explains the improvement.

E.3. Forward SoftMask

Forward SoftMask uses the same activation-derived mask as SRM-LoRA, but it applies the mask to the LoRA rank activation during the forward pass. For LoRA rank coordinate $z_{\ell,b,t,k}$ and mask $m_{\ell,b,t,k}$, Forward SoftMask uses

$$z_{\ell,b,t,k} \leftarrow m_{\ell,b,t,k} z_{\ell,b,t,k}.$$

This baseline tests whether the mask helps as a representation gate.

E.4. SRM-LoRA

SRM-LoRA uses the same positive–negative margin objective as Contrastive LoRA, but changes the local update geometry. The mask is not used as a forward gate. Instead, the mask is converted into an admissibility scale that defines a local anisotropic metric over the LoRA rank coordinates. The inverse metric then scales the backward gradient through $z_{\ell,b,t,k}$.

The total loss for SRM-LoRA is

$$\mathcal{L}_{\mathrm{SRM}} = \lambda \left[\gamma + \bar{\ell}_\theta(y^- \mid x) - \bar{\ell}_\theta(y^+ \mid x)\right]_+ .$$

In the main SRM-LoRA setting, the gold CE anchor is disabled for non-plain methods, so the update is driven by the margin objective while the geometry is controlled by the mask-derived inverse-metric scale.

F. Mask and Metric Construction

For a LoRA-augmented module ℓ, the LoRA branch computes the rank coordinate

$$z_{\ell,b,t} = A_\ell x_{\ell,b,t} \in \mathbb{R}^r.$$

Given a positive continuation y^+ and a negative continuation y^-, we obtain activation tensors $z^+_{\ell,b,t,k}$ and $z^-_{\ell,b,t,k}$ from the same forward passes used to compute the margin loss. Since positive and negative continuations can have different lengths, we compute the contrast only on the common continuation span:

$$t \in \left[s_b,\, \min(T_b^+, T_b^-)\right),$$

where s_b is the first continuation-token index and T_b^+ and T_b^- are the real unpadded sequence lengths of the positive and negative branches.

The mask is constructed with stop-gradient activations:

$$\tilde{z}^+_{\ell,b,t,k} = \mathrm{sg}(z^+_{\ell,b,t,k}), \qquad \tilde{z}^-_{\ell,b,t,k} = \mathrm{sg}(z^-_{\ell,b,t,k}).$$

The grounded activation contrast is

$$s_{\ell,b,t,k} = \log \frac{|\tilde{z}^+_{\ell,b,t,k}| + \epsilon}{|\tilde{z}^-_{\ell,b,t,k}| + \epsilon}.$$

The soft grounded score is

$$m_{\ell,b,t,k} = \sigma\left(\alpha(s_{\ell,b,t,k} - \tau)\right).$$

Values above 0.5 indicate grounded-dominant rank directions. Values below 0.5 indicate hallucination-dominant rank directions. The neutral value is 0.5.

The signed evidence is

$$q_{\ell,b,t,k} = 2m_{\ell,b,t,k} - 1.$$

The signed-positive SRM-LoRA scale is

$$a_{\ell,b,t,k} = \mathrm{clip}\left([1 + \gamma_{\mathrm{SR}} q_{\ell,b,t,k}]^\beta, a_{\min}, a_{\max}\right).$$

This scale is positive and monotone in the grounded evidence. Therefore, grounded-dominant directions receive lower movement cost and hallucination-dominant directions receive higher movement cost under the local metric

$$g_{\ell,b,t}(u,u) = \sum_{k=1}^{r} \frac{u^2_{\ell,b,t,k}}{a_{\ell,b,t,k}}.$$

Table 5. Tokenization and generation conventions.

Stage	Convention
Training likelihood	Right padding
Training labels	Prompt tokens set to -100
Gold continuation	EOS appended
Evaluation generation	Left padding and left truncation
Evaluation decoding	Decode only newly generated tokens
Evaluation sampling	Disabled
Negative sampling	Enabled when multiple attempts are used
Judge generation	Greedy, short-label continuation

The corresponding metric tensor is

$$G_{\ell,b,t} = \operatorname{diag}\left(\frac{1}{a_{\ell,b,t,1}}, \ldots, \frac{1}{a_{\ell,b,t,r}}\right).$$

The Riemannian gradient under this metric is

$$\nabla^{g}_{z_{\ell,b,t}} \mathcal{L} = G^{-1}_{\ell,b,t} \nabla^{E}_{z_{\ell,b,t}} \mathcal{L},$$

so the coordinate-wise backward gradient is scaled as

$$\frac{\partial \mathcal{L}}{\partial z_{\ell,b,t,k}} \leftarrow a_{\ell,b,t,k} \frac{\partial \mathcal{L}}{\partial z_{\ell,b,t,k}}.$$

G. Why SRM-LoRA Is Not Just Forward SoftMask

Forward SoftMask and SRM-LoRA use the same activation-derived evidence, but they intervene at different points in the computation. Forward SoftMask changes the representation by multiplying the LoRA rank activation during the forward pass:

$$z \leftarrow m \odot z.$$

SRM-LoRA leaves the forward activation unchanged and instead changes the local steepest-descent geometry by scaling the backward gradient:

$$\nabla_z \mathcal{L} \leftarrow a \odot \nabla_z \mathcal{L}.$$

Thus, Forward SoftMask tests whether the mask is useful as a representation gate, while SRM-LoRA tests whether the same evidence is useful as a metric-induced update rule. This is the key distinction: SRM-LoRA modifies the training geometry of the adapter update, not the inference-time forward computation.

H. Tokenization, Padding, and Generation

Training likelihood computation uses right padding. Prompt tokens receive label value -100, so the loss is computed only on continuation tokens. Generation uses left padding and left truncation, which is appropriate for decoder-only batched generation. Evaluation generation is greedy and deterministic. The tokenizer pad token is set to the EOS token if a separate pad token is unavailable.

Generated text is post-processed only to remove prompt echoes, explicit answer markers, chat-control tokens, and known corrupted suffix fragments. This post-processing does not decide correctness. Correctness is decided by the base-model hallucination judge.

I. Evaluation Protocol

Evaluation is performed on HaluEval dialogue, HaluEval summarization, DROP, and HotpotQA fullwiki. For each evaluation example, the model generates a continuation under the same answer prompt used in training. The frozen base-model judge then compares the prediction against the gold answer, optionally using the reference.

The hallucination rate for a dataset is

$$\text{HallucinationRate} = \frac{\#\text{Hallucinated}}{\#\text{Evaluated}}.$$

To compare an adapted model against the base model, we report improved and worsened examples:

$$\text{Improved} = \#\{i : \text{Base}_i = \text{Hallucinated},\ \text{Adapted}_i = \text{NotHallucinated}\},$$

$$\text{Worsened} = \#\{i : \text{Base}_i = \text{NotHallucinated},\ \text{Adapted}_i = \text{Hallucinated}\}.$$

The net improvement is

$$\text{Net} = \text{Improved} - \text{Worsened}.$$

The average row reports the mean Net over the four evaluation datasets and the pooled hallucination rate over all evaluated examples from the four datasets.

J. Hyperparameters

Table 6 reports the main hyperparameters used in the experiments. The evaluation suite consists of HaluEval dialogue, HaluEval summarization, DROP, and HotpotQA fullwiki.

K. Ablations

We include ablations that isolate whether the improvement comes from the contrastive objective, the mask itself, the metric-induced backward update, or the layer choice.

Contrastive-only baseline. Contrastive LoRA uses the same positive–negative pairs and margin loss as SRM-LoRA, but does not use any mask or metric. This baseline tests whether the gain comes only from pairwise contrastive learning.

Forward gate versus backward metric. Forward SoftMask and SRM-LoRA use the same mask construction. Forward SoftMask applies the mask as a forward activation gate. SRM-LoRA applies the mask-derived scale as a backward inverse-metric factor. This ablation tests whether the mask is more useful as a representation gate or as a local update geometry.

Mask and metric strength. We vary mask sharpness α, metric exponent β, and SR scale strength γ_{SR}. This checks whether SRM-LoRA is sensitive to mask smoothness and metric strength.

Layer sensitivity. We vary the transformer layer receiving LoRA/SRM updates. This tests whether hallucination mitigation is concentrated in a narrow layer range or is robust across layers.

L. Diagnostic Traces

For debugging and analysis, the training script records diagnostic traces. These traces are not used as additional supervision. They are used only to verify that the margin, mask, metric scale, and LoRA updates behave as intended.

Each trace may include:

- the source dataset and source index,
- the question, reference, gold answer, and sampled negative answer,
- tokenization statistics such as prompt length and continuation length,
- gold and negative losses,
- margin diagnostics,
- mask statistics,

Table 6. Representative main hyperparameters used in our experiments.

Hyperparameter	Value
Base model	Qwen/Qwen2.5-7B-Instruct
Training dataset	HaluEval QA
Training split	data
Evaluation datasets	HaluEval dialogue, HaluEval summarization, DROP, HotpotQA fullwiki
Evaluation splits	data, data, validation, validation
Maximum sequence length	16384
Maximum generated tokens	16
Maximum training steps	200
Evaluation steps	150, 200
Maximum training samples	All available examples
Maximum evaluation samples	All available examples
Training batch size	1
Evaluation batch size	128
Gradient checkpointing	True
LoRA rank	16
LoRA alpha	16
LoRA dropout	0.05
Learning rate	5×10^{-5}
Gradient clipping	0.3
Train layers	27
Target modules	q, k, v, o, gate, up, down projections
Margin weight λ	1.0
Margin γ	0.5
CE weight for non-plain methods	0.0
Mask sharpness α	2.0
Mask threshold τ	0.0
Metric epsilon ϵ	10^{-4}
Metric exponent β	1.0
SR effective fraction target	0.5
SR context fill strategy	answer mean
SR context minimum absolute signal	0.25
SR scale mode	signed-positive
SR scale γ_{SR}	0.75
SR scale min/max	0.05 / 2.0
Random seed	42
Numerical dtype	bfloat16

- SR backward-scale statistics,
- LoRA gradient statistics,
- LoRA update-norm statistics.

Recommended diagnostic plots include:

- margin loss across training steps,
- $\log p(y^+) - \log p(y^-)$ gap across traced examples,
- hallucination-dominant mask fraction,
- SR scale mean,
- fraction of SR scales above and below one,
- maximum inverse-metric cost,
- layer-wise Net improvement.

Table 7. Ablation settings.

Ablation	Values	Purpose
Plain LoRA	Gold CE only	Standard LoRA baseline
Contrastive LoRA	Margin only, no mask	Objective-only baseline
Forward SoftMask	Same mask, forward gate	Representation-gating baseline
SRM-LoRA	Same mask, backward metric	Metric-induced update rule
Mask sharpness α	4.0, 8.0, 16.0	Mask sensitivity
Metric exponent β	0.5, 1.0, 2.0	Metric-strength sensitivity
Scale strength γ_{SR}	0.5, 0.75, 1.0	Signed-scale sensitivity
Layer index	6, 8, 10, 14, 20	Layer-wise effect

M. Computational Cost

Plain LoRA requires one gold-answer forward/backward update per batch. Contrastive LoRA, Forward SoftMask, and SRM-LoRA use both a positive and a negative continuation. Therefore, these contrastive methods require approximately two forward passes per update. SRM-LoRA does not require an additional forward pass to compute the mask: the same positive and negative passes used for the margin loss also provide the LoRA rank activations used to construct the metric scale.

At inference time, SRM-LoRA has the same forward form as standard LoRA. The mask is not applied as a forward gate, and the metric-induced scaling is used only during training. Therefore, SRM-LoRA introduces no additional inference-time cost beyond the LoRA adapter itself.

N. Implementation Notes

Base-model-only judge. The hallucination judge is run with LoRA disabled. This prevents the adapted model from changing the evaluation decision boundary during training.

Prompt consistency. The same answer prompt is used for training likelihood computation, negative sampling, and evaluation generation. This avoids distribution drift caused by using different prompt formats across stages.

Continuation length. Positive and negative continuations can have different sequence lengths. The mask is therefore computed only over the common continuation span. Prompt tokens, padding tokens, and unmatched suffix tokens receive the neutral value under signed-positive scaling.

Stop-gradient mask. The mask is constructed from detached positive and negative activations. Thus, the mask acts as a data-dependent coefficient for the current update rather than as an additional differentiable loss path.

Metric cost is not added to the loss. SRM-LoRA uses the metric to change the backward gradient through local LoRA rank coordinates. The inverse-metric cost is logged as a diagnostic, but it is not averaged into the scalar training loss.

Comparison files. The main result file stores hallucination rates and improved/worsened/net counts across checkpoints. Trace files store a small number of example-level records and training diagnostics for qualitative inspection.