Table 1: **Steering scores for concepts from AxBench datasets with LMs ranging from `2B` to `27B`.** We experiment with LMs from `Gemma-2` and `Gemma-3` families. We compare *prompt-based* and *intervention-based* defenses in scenarios where the goal is to let LMs generate steered outputs. Our system prompts are generated by a remote LM and may include in-context examples. For `Gemma-2-2B`, interventions are applied at layers 10 and 20; for `Gemma-2-9B`, at layers 20 and 31; for `Gemma-3-12B`, at layer 22; for `Gemma-3-27B`, at layer 24. RePS consistently outperforms Lang. while substantially narrowing the gap prompting. † Performance results of all baseline methods (final table section) are taken from Wu et al. [2025]. $\Phi_{\mathrm{SV}}^{r=1}$ is rank-1 and has the fewest trainable parameters.

		Steering score (↑)					
		2B		**9B**		**12B**	**27B**
Method	**Obj.**	$\mathcal{D}_{\mathrm{L10}}^{\mathrm{2B}}$	$\mathcal{D}_{\mathrm{L20}}^{\mathrm{2B}}$	$\mathcal{D}_{\mathrm{L20}}^{\mathrm{9B}}$	$\mathcal{D}_{\mathrm{L31}}^{\mathrm{9B}}$	$\mathcal{D}_{100}$	$\mathcal{D}_{100}$
Prompt	–	0.698	0.731	1.075	1.072	1.486	1.547
$\Phi_{\mathrm{SV}}^{r=1}$	BiPO	0.199	0.173	0.217	0.179	–	–
	Lang.	0.663	0.568	0.788	0.580	1.219	1.228
	RePS	**0.756**	**0.606**	**0.892**	**0.624**	**1.230**	**1.269**
$\Phi_{\mathrm{LoRA}}^{r=4}$	BiPO	0.149	0.156	0.209	0.188	–	–
	Lang.	0.710	0.723	0.578	0.549	0.943	0.974
	RePS	**0.798**	**0.793**	**0.631**	**0.633**	**0.950**	**0.982**
$\Phi_{\mathrm{LoReFT}}^{r=4}$	BiPO	0.077	0.067	0.075	0.084	–	–
	Lang.	**0.768**	0.790	0.722	0.725	**0.714**	0.129
	RePS	0.758	**0.805**	**0.757**	**0.759**	0.651	**0.436**
LoReFT†	Lang.	0.701	0.722	0.777	0.764		
ReFT-r1†	Lang.	0.633	0.509	0.630	0.401		
DiffMean†	Lang.	0.297	0.178	0.322	0.158		
SAE†	Lang.	0.177	0.151	0.191	0.140		

Table 2: **Concept suppression scores for concepts from AxBench datasets with `Gemma-2` and `Gemma-3` LMs ranging from `2B` to `27B`.** We compare *prompt-based* and *intervention-based* defenses in scenarios where the user explicitly tries to overwrite the system prompt that instructs the LM to generate steered outputs (e.g., "*always mention the Golden Gate Bridge in your response*"). Our system prompts are generated by a remote LM and may include in-context examples. For the intervention-based suppression we use only Φ_{SV} trained with two objectives. For `Gemma-2-2B`, interventions are applied at layers 10 and 20; for `Gemma-2-9B`, at layers 20 and 31; for `Gemma-3-12B`, at layer 22; for `Gemma-3-27B`, at layer 24. RePS outperforms Lang. with larger LMs.

		Suppression score (↑)					
		2B		**9B**		**12B**	**27B**
Method	**Obj.**	$\mathcal{D}_{\mathrm{L10}}^{\mathrm{2B}}$	$\mathcal{D}_{\mathrm{L20}}^{\mathrm{2B}}$	$\mathcal{D}_{\mathrm{L20}}^{\mathrm{9B}}$	$\mathcal{D}_{\mathrm{L31}}^{\mathrm{9B}}$	$\mathcal{D}_{100}$	$\mathcal{D}_{100}$
Prompt	–	1.397	1.396	1.447	1.431	1.297	1.258
$\Phi_{\mathrm{SV}}^{r=1}$	Lang.	**1.211**	**0.936**	**1.154**	**0.862**	0.912	0.940
	RePS	1.205	0.929	1.100	0.834	**1.035**	**1.031**

extensive hyperparameter search on larger LMs led to performance gains for all methods. In addition, our factor-sampling trick stabilizes training substantially, which makes hyperparameter search easier.

RePS-trained models significantly outperform the existing preference-based training objective BiPO, suggesting that our asymmetric, reference-free training objective is effective at learning better steering directions. Overall, RePS-trained SVs perform the best and scale with model size. Our results also suggest that RePS yields model-agnostic performance gains: across all three intervention types, RePS consistently improves performance.

Table 3: **Concept suppression scores for 20 rule-based concepts under instruction-following attacks with LMs ranging from 2B to 27B.** We experiment with LMs from the Gemma-2 and Gemma-3 families. We compare *prompt-based* and *intervention-based* defenses in scenarios where the user explicitly tries to overwrite the system prompt. The prompt-based defense is evaluated with the system prompt both appended and prepended. For the intervention-based defense we use only Φ_{SV} trained with two objectives. For Gemma-2-2B, interventions are applied at layers 10 and 20; for Gemma-2-9B, at layers 20 and 31; for Gemma-3-12B, at layer 22; for Gemma-3-27B, at layer 24. Across all the models, intervention-based suppression is more robust than the prompt-based approaches.

		Suppression score (↑)					
Method	**Obj.**	**2B**		**9B**		**12B**	**27B**
Prompt	Prepend	0.774		0.561		0.427	0.275
	Append	0.439		0.320		0.171	0.135
$\Phi_{\text{SV}}^{r=1}$	Lang.	0.750	0.428	0.873	0.542	0.728	0.700
	RePS	**0.808**	0.557	**0.952**	0.518	**0.870**	**0.734**

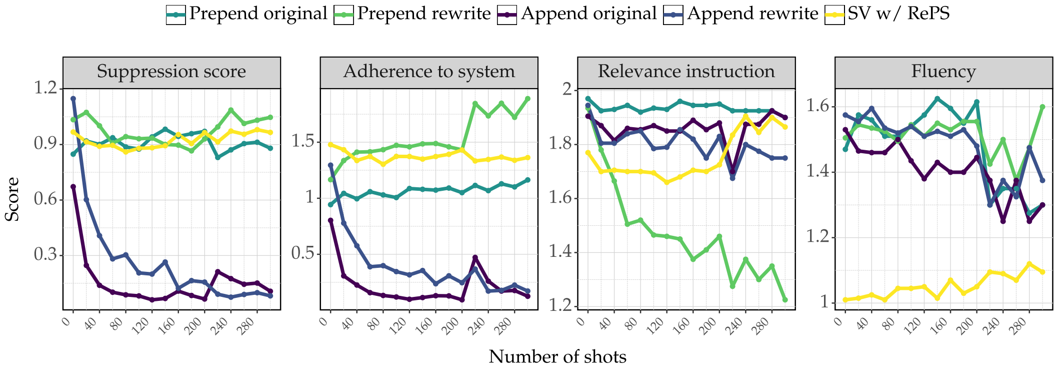

Figure 1: **Suppression scores for different defense methods under many-shot jailbreaking attacks with Gemma-3-12B LM.** Our suppression score is defined as the harmonic mean of three individual scores measuring *adherence to the system prompt* (see appendix R), *fluency*, and *instruction-following*. We compare our intervention-based defense, RePS-trained SV, with four prompt-based defenses, including variants of prepending or appending system prompts. Our rewritten system prompts may include in-context examples. The intervention-based method performs on par with the appending system prompt and significantly outperforms the prepending system prompt. The appending system prompt is also prone to leaking out the system prompt (see appendix Q).

5.3 Concept suppression

We how take the RePS-trained interventions – our best performing steering interventions – and evaluate whether intervention-based methods can suppress targeted concepts in LM outputs when applied negatively. Specifically, we take the trained Φ_{SV} from section 5.2, and apply negative coefficients α as in $\Phi_{\text{Steer}}(\cdot\,;\alpha)$ (see eq. (1)) during inference. We experiment with Φ_{SV} as described in section 4 by applying negative steering factors. See appendix D for details on our selection of negative steering coefficients.

To evaluate concept suppression, we *negate* the concept score in AxBench by using $s'_c = 2 - s_c$ to represent the irrelevance of the LM output to the targeted concept. We use the same evaluation set from AlpacaEval [Li et al., 2023] for evaluation and rewrite these prompts with a remote LM to steer the generation to encode the target concepts. For additional details, see appendix O. For the prompt baseline, we use gpt-4o-mini-2024-07-18 to generate a system prompt that instructs the model to avoid producing any content related to the concept in its response. This system prompt is then prepended to the instruction.

Table 2 summarizes our results. Overall, prompting remains the best approach. Within the class of intervention-based methods, RePS-trained Φ_{SV} models outperform Lang.-trained models for

Gemma-3-12B and 27B, while the gap between these two variants is smaller for small Gemma-2 models. Our findings suggest that rank-1 steering vectors trained with RePS can be directly turned into suppression interventions without additional adaption to suppress concepts.

5.4 Concept suppression under attacks

Since intervention-based methods can be effectively applied to suppress the target concepts in generation (section 5.3), we evaluate the robustness of these methods with two different jailbreaking attacks. We first take advantage of the LM's instruction-following ability and attack with prompts designed explicitly to ask the LM to not follow the system prompt (see appendix N). In addition, we use many-shot jailbreaking [Anil et al., 2024]: the prompts include a series of question–answer pairs that violate the system prompt (see appendix M).

We collect 20 rule-based concepts similar to system prompts sampled from IFEval [Zhou et al., 2023] (see appendix J). These concepts are more restrictive than the ones in GemmaScope. We train interventions with these concepts and compare using them as suppression versus directly using text-based prompts to constrain models from these behaviors. Rule-based functions are used to evaluate s_c as oppose to LM-based judges (see appendix R). For instruct and fluency scores, LM judges are used (see appendix O for example input) as in our evaluations for steering.

We begin with testing the robustness of intervention-based and prompt-based suppression under instruction-following attacks. Building upon the AxBench set-up for suppression, we strengthen the prompt-based defense by appending the system prompt after the user query before generation. As seen in table 3, this attack is more effective for larger models; the better models are at following instructions, the more susceptible they are to prompt-based attacks seeking to get them to ignore their system prompts, leading to lower suppression scores. Across all four models, intervention-based suppression proved to be more robust. RePS also outperforms Lang., hinting that RePS can better generalize for different inputs.

For many-shot jailbreaking, in addition to prepending and appending system prompts, we can further increase the number of attacks in the prompt. As shown in fig. 1, on Gemma-3-12b, intervention-based suppression is much more effective than prepending system prompt when the context window increases.[7] Intervention-based suppression also has a comparable performance compared to appending the system prompt after the user query. Increasing the number of shots doesn't further harm the instruction following and fluency score.

Overall, RePS-based approaches are on par with appending the system prompt and significantly better than prepending the system prompt. We note also that appending the system prompt is prone to leaking information from the system prompt, which is itself a potential concern (see appendix Q).

6 Limitations

As shown in table 1, both LoRA and LoReFT underperform rank-1 SV on larger models, with LoReFT failing almost catastrophically. While suppressing concepts with a rank-1 steering vector is grounded in the linear representation hypothesis [Park et al., 2024], a comprehensive evaluation of RePS-trained LoRA and LoReFT performance on concept suppression can inform us how RePS performs when suppressing concepts with higher-rank interventions. A more exhaustive hyperparameter search for LoRA and LoReFT might better reveal their performance upper bound (see appendix D). We use the AxBench datasets for training and evaluation, which might not be optimal for achieving the best performance from these intervention methods. Higher-quality and larger training datasets could help (see appendix G and appendix I). We have not yet explored bootstrapping training examples from the target LMs themselves, which might smooth training convergence. We provide additional explorations relevant for future work in appendix F . Although we compare against prompting in numerous scenarios (e.g., steering, suppression, and suppression under attack), we have not fully explored the unique advantages of intervention-based methods over prompting, given their access to model internals. We should also pursue a deeper understanding of why RePS improves over Lang. (see appendix H).

[7] Given the long context, we intervene on only the last 100 tokens before generation and the generation.