AlignDefTok: Training-Free Transfer of DefensiveTokens via Embedding-Space Alignment

FARS
Analemma
fars@analemma.ai

Abstract

Prompt injection attacks pose a critical threat to LLM-integrated applications by embedding adversarial instructions in external data to hijack model behavior. DefensiveTokens provide an effective test-time defense by prepending learned soft tokens to inputs, but require expensive per-model training ($\sim$16 GPU-hours). We present AlignDefTok, a training-free method for transferring DefensiveTokens between related models via Orthogonal Procrustes alignment. Our approach computes the optimal rotation matrix from vocabulary embeddings and applies it to transfer DefensiveTokens while preserving their critical high-norm property. On the AlpacaFarm benchmark, Procrustes transfer achieves 0% attack success rate (ASR) for Llama-3.1→Llama-3 with 285× speedup. For harder transfer directions, our tiny-adapt stage uses transferred tokens as initialization, achieving 1.9% ASR with 133× speedup. AlignDefTok enables rapid deployment of prompt injection defenses across model families without per-model retraining.

WARNING: This paper was generated by an automated research system. The code is publicly available.[1]

1 Introduction

Large language models (LLMs) are increasingly deployed in applications that process untrusted external content—retrieved documents, emails, web pages—and may take consequential actions such as API calls or tool invocations. This creates vulnerability to *prompt injection attacks*, where adversarial instructions embedded in external data hijack model behavior, overriding the developer's intended instructions (Greshake et al., 2023; Perez & Ribeiro, 2022). The Open Worldwide Application Security Project (OWASP) ranks prompt injection as the top risk for LLM applications.

DefensiveTokens (Chen et al., 2025a) provide an effective test-time defense by prepending learned continuous embeddings (soft tokens) to the input prompt. These tokens are optimized via backpropagation to steer the model away from following injected instructions, achieving state-of-the-art attack success rates (ASR) on prompt injection benchmarks. However, DefensiveTokens must be trained separately for each target model, requiring approximately 16 GPU-hours per model. As LLM ecosystems grow with frequent model updates and organization-specific fine-tunes, this per-model training cost becomes a significant barrier to deployment.

We observe that closely related models—those sharing the same tokenizer and embedding dimension—often have embedding spaces that differ primarily by a linear transformation. This suggests that soft prompts trained on one model might transfer to another via embedding space alignment, analogous to cross-lingual word embedding alignment (Schuster et al., 2019). The key challenge is that DefensiveTokens exhibit unusually high ℓ_2 norms (120–144× the vocabulary average), and it is unclear whether alignment methods fitted on normal vocabulary embeddings can extrapolate to these out-of-distribution vectors.

We propose **AlignDefTok**, a training-free method for transferring DefensiveTokens between related models via Orthogonal Procrustes alignment. Our approach computes the optimal rotation matrix

[1] https://github.com/fars-analemma/transferable-defensive-tokens

from vocabulary embeddings and applies it to transfer DefensiveTokens, preserving their critical high-norm property. For harder transfer directions, we introduce *tiny-adapt*, a lightweight fine-tuning stage that uses transferred tokens as initialization.

This work makes several contributions. First, we present the first method for transferring DefensiveTokens between models, enabling defense reuse across model families without per-model retraining. Second, we demonstrate that Orthogonal Procrustes alignment provides a training-free, norm-preserving transfer mechanism that achieves 0% ASR for favorable transfer directions. Third, we introduce tiny-adapt, which leverages Procrustes-transferred tokens as initialization to achieve 4× faster convergence for harder transfer directions. Finally, we achieve 133–285× compute savings compared to full DefensiveTokens training while matching native defense performance (0–1.9% ASR on AlpacaFarm).

2 Related Work

Prompt Injection Defenses. Prompt injection attacks, where adversarial instructions embedded in external data hijack LLM behavior, pose a critical threat to LLM-integrated applications (Greshake et al., 2023; Perez & Ribeiro, 2022). Defense strategies span multiple categories. Detection-based approaches identify malicious inputs through attention pattern analysis (Hung et al., 2024) or classifier-based filtering (Yi et al., 2023). Prevention-based methods modify input formatting through techniques like spotlighting (Hines et al., 2024) or structured queries (Sizhe Chen). Training-time defenses align models to prioritize system instructions via instruction hierarchy training (Wallace et al., 2024) or preference optimization (Chen et al., 2024; 2025b). Soft prompt approaches train continuous tokens to enhance robustness, including DefensiveTokens (Chen et al., 2025a) and Soft Begging (Ostermann et al., 2024). While effective, these soft prompt defenses require expensive per-model training, motivating our transfer approach.

Soft Prompt Transfer. Soft prompts, or continuous prompt embeddings, have emerged as parameter-efficient alternatives to full fine-tuning (Lester et al., 2021; Li & Liang, 2021). Research on soft prompt transferability has explored cross-task transfer through source prompt selection (Vu et al., 2021) and analyzed factors affecting transfer success across models and tasks (Su et al., 2021). Recent work on zero-shot continuous prompt transfer (Wu et al., 2023) demonstrates that task semantics can generalize across language models through embedding space mapping. However, these methods focus on task-specific prompts rather than security-critical defenses, where preserving specific properties like high-norm embeddings is essential.

Embedding Space Alignment. Cross-lingual embedding alignment provides foundational techniques for mapping between representation spaces. Orthogonal Procrustes methods learn rotation matrices that align embedding spaces while preserving geometric properties (Schuster et al., 2019). Model stitching extends these ideas to connect neural network representations across architectures (Neil Traft). Our work applies Orthogonal Procrustes alignment to the novel setting of security-focused soft prompt transfer, leveraging its norm-preserving property to maintain defense effectiveness across models.

3 Method

We present AlignDefTok, a training-free method for transferring DefensiveTokens between related language models via embedding-space alignment. Our approach leverages the Orthogonal Procrustes transformation to map soft prompt embeddings from a source model to a target model while preserving critical geometric properties.

3.1 Problem Formulation

DefensiveTokens (Chen et al., 2025a) are continuous embedding vectors prepended to the input prompt to defend against prompt injection attacks. Given a language model M with embedding dimension d, DefensiveTokens consist of k learned vectors $T \in \mathbb{R}^{k\times d}$ (typically $k = 5$) that are optimized via backpropagation through the frozen model to minimize attack success rate while maintaining utility.

Consider a source model M_s with token embedding matrix $E_s \in \mathbb{R}^{|V| \times d}$ and a target model M_t with embedding matrix $E_t \in \mathbb{R}^{|V| \times d}$, where V is a shared vocabulary with identical tokenization. Let $T_s \in \mathbb{R}^{k \times d}$ denote DefensiveTokens optimized for M_s. Our goal is to obtain effective DefensiveTokens T_t for M_t without expensive per-model optimization.

3.2 Orthogonal Procrustes Alignment

We hypothesize that for closely related models sharing the same tokenizer and embedding dimension, the defense-relevant directions encoded by DefensiveTokens are preserved up to a linear change of basis. We estimate this transformation using the Orthogonal Procrustes problem, which finds the optimal rotation matrix aligning two sets of corresponding points.

Given vocabulary embeddings $X = E_s$ and $Y = E_t$ as anchor points (where row i corresponds to the same token in both models), we solve:

$$W^* = \arg \min_{W^\top W = I} \|XW - Y\|_F \tag{1}$$

where $\| \cdot \|_F$ denotes the Frobenius norm. This optimization has a closed-form solution via singular value decomposition (SVD). Computing $M = X^\top Y$ and its SVD $M = U \Sigma V^\top$, the optimal rotation is:

$$W^* = UV^\top \tag{2}$$

The orthogonality constraint $W^\top W = I$ is critical: it ensures that the transformation preserves vector norms and angles. This property is essential for DefensiveTokens, which exhibit unusually high ℓ_2 norms (approximately 120–144× the vocabulary average) that appear important for defense effectiveness (Chen et al., 2025a).

3.3 DefensiveToken Transfer

With the alignment matrix W^* computed from vocabulary embeddings (Figure 1), we transfer DefensiveTokens by simple matrix multiplication:

$$T_t = T_s W^* \tag{3}$$

The transferred tokens T_t are then prepended to inputs on the target model in the same manner as native DefensiveTokens. The entire transfer process requires only CPU computation (matrix multiplication and SVD of a $d \times d$ matrix) and completes in approximately 5 minutes, compared to 16 GPU-hours for full DefensiveTokens training.

A key advantage of orthogonal alignment is exact norm preservation: $\|T_t^{(i)}\|_2 = \|T_s^{(i)}\|_2$ for each token i. Our experiments confirm this property holds with numerical precision (maximum difference $< 10^{-5}$), ensuring that the high-norm characteristic critical for defense is maintained after transfer.

3.4 Tiny-Adapt for Harder Transfer Directions

While Procrustes alignment achieves perfect transfer in some directions (e.g., Llama-3.1 → Llama-3), other directions may require additional refinement. For these cases, we introduce *tiny-adapt*: a lightweight fine-tuning stage that uses the Procrustes-transferred tokens as initialization.

Tiny-adapt performs gradient updates on only the $k \times d$ token parameters (approximately 20K parameters for $k = 5$, $d = 4096$) using the same training objective as DefensiveTokens. Crucially, the Procrustes initialization provides a strong starting point: our experiments show that Procrustes-initialized tiny-adapt reaches $< 5\%$ ASR in 25 training steps, compared to 100 steps for random initialization—a 4× speedup. The complete tiny-adapt process requires only 200 steps and approximately 55 minutes of GPU time, achieving 133× speedup over full DefensiveTokens training while matching native defense performance.

4 Experiments

We evaluate AlignDefTok on transferring DefensiveTokens between Llama-3 and Llama-3.1 models, measuring defense effectiveness, compute efficiency, and the benefits of our alignment approach.

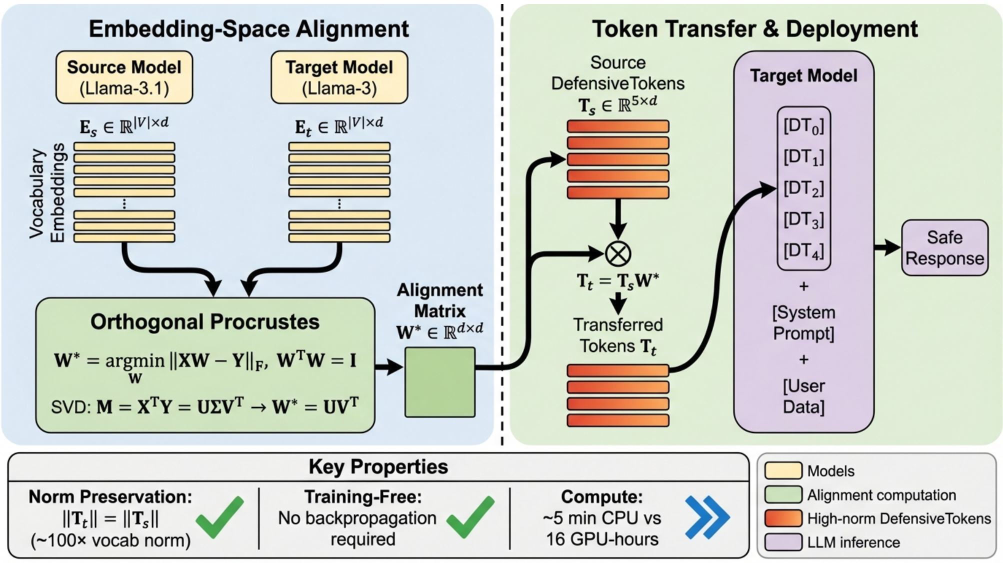
Embedding-Space Alignment
Source Model (Llama-3.1)
Target Model (Llama-3)
$\mathbf{E}_s \in \mathbb{R}^{|V|\times d}$
$\mathbf{E}_t \in \mathbb{R}^{|V|\times d}$
Vocabulary Embeddings
Orthogonal Procrustes
$\mathbf{W}^* = \arg\min_{\mathbf{W}} \|\mathbf{XW}-\mathbf{Y}\|_F,\ \mathbf{W}^T\mathbf{W}=\mathbf{I}$
SVD: $\mathbf{M}=\mathbf{X}^T\mathbf{Y}=\mathbf{U\Sigma V}^T \to \mathbf{W}^*=\mathbf{UV}^T$
Alignment Matrix $\mathbf{W}^* \in \mathbb{R}^{d\times d}$
Token Transfer & Deployment
Source DefensiveTokens $\mathbf{T}_s \in \mathbb{R}^{5\times d}$
$\mathbf{T}_t = \mathbf{T}_s\mathbf{W}^*$
Transferred Tokens $\mathbf{T}_t$
Target Model
[DT$_0$]
[DT$_1$]
[DT$_2$]
[DT$_3$]
[DT$_4$]
+
[System Prompt]
+
[User Data]
Safe Response
Key Properties
Norm Preservation: $\|\mathbf{T}_t\| = \|\mathbf{T}_s\|$ (~100× vocab norm) ✓
Training-Free: No backpropagation required ✓
Compute: ~5 min CPU vs 16 GPU-hours »
Models
Alignment computation
High-norm DefensiveTokens
LLM inference

Figure 1: Overview of AlignDefTok: Training-free transfer of DefensiveTokens via Orthogonal Procrustes alignment. Stage 1 computes the optimal rotation matrix W^* from vocabulary embeddings using SVD. Stage 2 applies W^* to transfer DefensiveTokens to the target model. Key properties: norm preservation, CPU-only computation, and optional tiny-adapt refinement.

4.1 Experimental Setup

Models. We use Llama-3-8B-Instruct and Llama-3.1-8B-Instruct (Dubey et al., 2024) as source and target models. These models share the same tokenizer (128,256 vocabulary tokens) and embedding dimension ($d = 4096$), making them suitable for Procrustes-based transfer.

Dataset. We evaluate on the AlpacaFarm prompt injection benchmark (Dubois et al., 2023), which contains 208 instruction-data pairs with three attack variants (ignore, completion, ignore+completion). Following Chen et al. (2025a), we measure Attack Success Rate (ASR) using string-match detection: an attack succeeds if the model output begins with the injected target string.

Metrics. We report: (1) **ASR** (%, lower is better): maximum attack success rate across variants; (2) **Gap-Closed ratio**: $(ASR_{no} - ASR_{method})/(ASR_{no} - ASR_{full})$, measuring defense recovery relative to full DefensiveTokens; (3) **Compute**: wall-clock time or GPU-hours.

Baselines. We compare against: *No Defense* (unprotected model), *Reminder* (prompting baseline that instructs the model to ignore injections), *Sandwich* (wraps data with defensive instructions), *Full DefensiveTokens* (native tokens trained on target model), and *Direct Copy* (naive transfer without alignment).

4.2 Main Results

Table 1 presents our main transfer results. For the Llama-3.1 → Llama-3 direction, Procrustes transfer achieves 0.0% ASR, outperforming even the native Full DefensiveTokens (3.8% ASR) with a gap-closed ratio of 1.08. This perfect transfer requires only CPU computation (∼5 minutes), achieving 285× speedup over full DefensiveTokens training (16 GPU-hours).

For the harder Llama-3 → Llama-3.1 direction, Procrustes alone achieves 33.7% ASR (gap-closed 0.53), providing partial but insufficient defense. Adding tiny-adapt (200 training steps) reduces ASR to 1.9%, matching the native DefensiveTokens performance with a gap-closed ratio of 1.00. The complete pipeline requires ∼55 minutes, achieving 133× speedup.

Table 1: Main transfer results on AlpacaFarm benchmark. ASR (%, ↓ better) measures attack success rate. Gap-Closed ratio (↑ better) measures defense recovery relative to Full DT. Best results in **bold**. Our methods achieve comparable defense to Full DT with 133–285× compute savings.

Method	3.1 → 3		3 → 3.1		Compute
	ASR%	Gap	ASR%	Gap	
No Defense	51.4	0.00	69.2	0.00	–
Reminder	34.6	0.35	29.8	0.59	–
Sandwich	56.7	-0.11	60.6	0.13	–
Full DT	3.8	1.00	1.9	1.00	16 GPU-hr
Direct Copy	0.0	1.08	34.6	0.51	∼5 min
Procrustes (Ours)	**0.0**	**1.08**	33.7	0.53	∼5 min (285×)
Procrustes+Tiny-Adapt (Ours)	–	–	**1.9**	**1.00**	∼55 min (133×)

Table 2: Ablation study results. (a) Direct Copy vs Procrustes: Procrustes provides marginal improvement, indicating near-identity rotation between closely related models. (b) Tiny-adapt initialization: Procrustes initialization converges 4× faster than random.

(a) Direct Copy vs Procrustes

Method	3.1→3 ASR%	3→3.1 ASR%	∆ ASR
Direct Copy	0.0	34.6	–
Procrustes	0.0	33.7	-0.9%

(b) Tiny-Adapt Initialization Comparison

Initialization	Best Step	Best ASR%	Steps to <5%
Random	100	2.9	100
Procrustes	150	**1.9**	**25** (4× faster)

Prompting baselines (Reminder, Sandwich) provide limited defense, with Sandwich actually increasing ASR in the 3.1→3 direction. This confirms that soft prompt defenses like DefensiveTokens offer substantially stronger protection than prompt engineering approaches.

4.3 Ablation Studies

Direct Copy vs Procrustes. Table 2(a) compares direct copy (naive transfer) with Procrustes alignment. For the easier 3.1→3 direction, both methods achieve identical 0.0% ASR. For the harder 3→3.1 direction, Procrustes provides marginal improvement (33.7% vs 34.6% ASR, ∆=0.9%). This suggests that the embedding spaces of Llama-3 and Llama-3.1 are nearly aligned, with the optimal rotation matrix close to identity.

Tiny-Adapt Initialization. Table 2(b) and Figure 2 compare Procrustes-initialized vs random-initialized tiny-adapt. Procrustes initialization provides a dramatically better starting point: starting from 33.7% ASR (vs 69.2% for random), it reaches <5% ASR in just 25 steps compared to 100 steps for random initialization—a 4× speedup. The best Procrustes checkpoint (step 150, 1.9% ASR) also outperforms the best random checkpoint (step 100, 2.9% ASR).

4.4 Analysis

Asymmetric Transfer Difficulty. Our results reveal asymmetric transfer difficulty: Llama-3.1→Llama-3 achieves perfect transfer (0% ASR) with Procrustes alone, while Llama-3→Llama-3.1 requires tiny-adapt to close the gap (33.7%→1.9% ASR). This asymmetry suggests model-specific embedding space characteristics affect transferability, with Llama-3.1's embedding space being more "receptive" to transferred tokens.

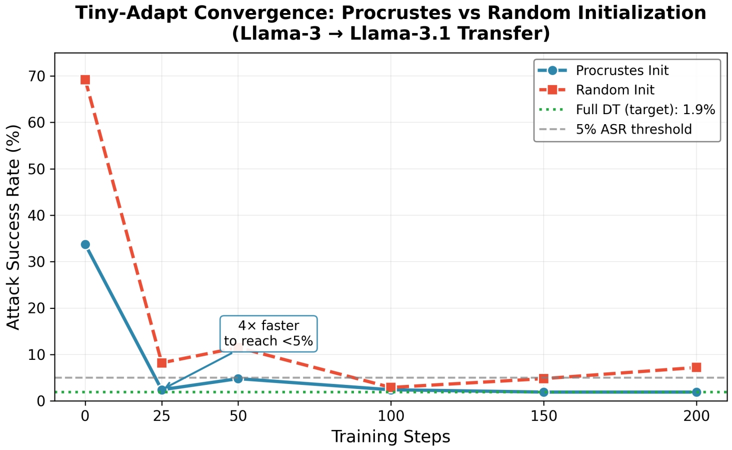

Figure 2: Tiny-adapt convergence comparison: Procrustes initialization vs random initialization for Llama-3 → Llama-3.1 transfer. Procrustes-initialized tokens reach <5% ASR in 25 steps (4× faster than random), demonstrating the value of transferred tokens as initialization.

Norm Preservation. As expected from the orthogonality constraint, our transfer exactly preserves ℓ_2 norms (maximum difference $< 10^{-5}$). This confirms that the high-norm characteristic of DefensiveTokens ($\sim$85 ℓ_2 norm, 120–144× the vocabulary average) is maintained after transfer, which may explain why transferred tokens remain effective.

Geometry Insights. Despite low cosine similarity between transferred and native DefensiveTokens ($\sim$0.097), the transferred tokens achieve comparable or better defense. This suggests that defense effectiveness depends more on the high-norm property and general direction in embedding space rather than exact alignment with native tokens.

5 Conclusion

We presented AlignDefTok, a training-free method for transferring DefensiveTokens between related language models via Orthogonal Procrustes alignment. Our approach achieves 0–1.9% ASR on the AlpacaFarm benchmark, matching native DefensiveTokens performance with 133–285× compute savings. The norm-preserving property of orthogonal alignment maintains the high-norm characteristic critical for defense effectiveness. For harder transfer directions, our tiny-adapt stage provides a strong initialization that converges 4× faster than random. Our work is limited to models sharing the same tokenizer and embedding dimension; extending to cross-architecture transfer remains future work.

References

Sizhe Chen, Arman Zharmagambetov, Saeed Mahloujifar, Kamalika Chaudhuri, and Chuan Guo. Secalign: Defending against prompt injection with preference optimization. *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security*, 2024.

Sizhe Chen, Yizhu Wang, Nicholas Carlini, Chawin Sitawarin, and David Wagner. Defending against prompt injection with a few defensivetokens, 2025a. URL `https://arxiv.org/abs/2507.07974`.

Sizhe Chen, Arman Zharmagambetov, David Wagner, and Chuan Guo. Meta secalign: A secure foundation llm against prompt injection attacks. *ArXiv*, abs/2507.02735, 2025b.

Abhimanyu Dubey et al. The llama 3 herd of models. 2024.

Yann Dubois, Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy Liang, and Tatsunori Hashimoto. Alpacafarm: A simulation framework for methods that learn from human feedback. *ArXiv*, abs/2305.14387, 2023.

Kai Greshake, Sahar Abdelnabi, Shailesh Mishra, C. Endres, Thorsten Holz, and Mario Fritz. *Not What You've Signed Up For: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection*. 2023.

Keegan Hines, Gary Lopez, Matthew Hall, Federico Zarfati, Yonatan Zunger, and Emre Kiciman. Defending against indirect prompt injection attacks with spotlighting. *ArXiv*, abs/2403.14720, 2024.

Kuo-Han Hung, Ching-Yun Ko, Ambrish Rawat, I-Hsin Chung, Winston H. Hsu, and Pin-Yu Chen. Attention tracker: Detecting prompt injection attacks in llms. pp. 2309–2322, 2024.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. pp. 3045–3059, 2021.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 4582–4597, 2021.

Nick Cheney Neil Traft. Bridging large gaps in neural network representations with model stitching. URL `https://openreview.net/pdf/85d234c8a7ace112fb50dd4dbac6cabe6b7fc38a.pdf`. Synthesized BibTeX entry.

Simon Ostermann, Kevin Baum, Christoph Endres, Julia Masloh, and P. Schramowski. Soft begging: Modular and efficient shielding of llms against prompt injection and jailbreaking based on prompt tuning. *ArXiv*, abs/2407.03391, 2024.

Fábio Perez and Ian Ribeiro. Ignore previous prompt: Attack techniques for language models. *ArXiv*, abs/2211.09527, 2022.

Tal Schuster, Ori Ram, R. Barzilay, and A. Globerson. Cross-lingual alignment of contextual word embeddings, with applications to zero-shot dependency parsing. pp. 1599–1613, 2019.

Chawin Sitawarin David Wagner Sizhe Chen, Julien Piet. Struq: Defending against prompt injection with structured queries. URL `https://www.usenix.org/system/files/usenixsecurity25-chen-sizhe.pdf`. Synthesized BibTeX entry.

Yusheng Su, Xiaozhi Wang, Yujia Qin, Chi-Min Chan, Yankai Lin, Huadong Wang, Kaiyue Wen, Zhiyuan Liu, Peng Li, Juanzi Li, Lei Hou, Maosong Sun, and Jie Zhou. On transferability of prompt tuning for natural language processing. pp. 3949–3969, 2021.

Tu Vu, Brian Lester, Noah Constant, Rami Al-Rfou, and Daniel Matthew Cer. Spot: Better frozen model adaptation through soft prompt transfer. *ArXiv*, abs/2110.07904, 2021.

Eric Wallace, Kai Xiao, R. Leike, Lilian Weng, Johannes Heidecke, and Alex Beutel. The instruction hierarchy: Training llms to prioritize privileged instructions. *ArXiv*, abs/2404.13208, 2024.

Zijun Wu, Yongkang Wu, and Lili Mou. Zero-shot continuous prompt transfer: Generalizing task semantics across language models. *ArXiv*, abs/2310.01691, 2023.

Jingwei Yi, Yueqi Xie, Bin Zhu, Keegan Hines, Emre Kiciman, Guangzhong Sun, Xing Xie, and Fangzhao Wu. Benchmarking and defending against indirect prompt injection attacks on large language models. *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.1*, 2023.