Towards Plug-and-Play Attribute Control Modules in Text Generation: An Exploratory Study of LoRA Portability Across Model Families

Michela Lorandi and Anya Belz
ADAPT, Dublin City University
Dublin, Ireland
michela.lorandi2@mail.dcu.ie anya.belz@dcu.ie

Abstract

Parameter-efficient fine-tuning (PEFT) methods, such as LoRA, produce structurally modular parameter sets. However, whether this structural modularity also implies *functional* modularity, and whether task-specific control moreover transfers to a different host model without retraining, remains largely untested for controlled text generation (CTG). We present an exploratory case study on the cross-model portability of attribute-specialised LoRA modules trained for sentiment and topic control. Adapting the LoRASuite framework (Li et al., 2025), we port modules across four host models spanning three conditions of increasing difficulty: same family/different size, different family/comparable size, and different family/different size. We evaluate these in (i) a zero-shot regime, where the ported module is used without further training; and (ii) a few-step regime, where a small amount of target-side fine-tuning follows transfer. Our results show that zero-shot porting is in most cases not sufficient to preserve control effectiveness, ported modules scoring not much better than the untuned base model. However, we show that a short post-porting fine-tuning step recovers a substantial part of the gap to the non-ported upper bound, suggesting that algebraic transfer is best understood as a strong initialisation for lightweight adaptation rather than as a fully training-free plug-and-play mechanism.

1 Introduction

Parameter-efficient fine-tuning (PEFT) methods such as LoRA (Hu et al., 2022) have become the default mechanism for adapting large language models (LLMs) to controlled text generation (CTG) tasks, allowing a single frozen base model to be specialised for a target attribute (e.g. sentiment, topic) via a lightweight set of trainable parameters. Because these modules interact with the host model through a well-defined interface (attention and feed-forward projections), they are often characterised as *structurally modular*: a self-contained unit that can, in principle, be attached to a different host model. Whether this structural modularity also implies *functional modularity*, that is, whether the task-specific control learned by the module survives transplantation to a new base model without retraining, remains an open empirical question, and one with considerable practical weight. Base models are updated or superseded at a rapid pace, and each new release conventionally requires attribute-specialised modules to be retrained from scratch, at a training and data cost that scales with the number of attributes and base models an organisation wishes to support. If task-specific control can instead be ported algebraically between hosts, the useful lifetime of a trained module could be decoupled from that of any single base model.

Prior work offers partial answers. Sabry and Belz (2024) provide the first large-scale evidence that PEFT modules are portable between same-family hosts, though restricted to general-purpose classification tasks. More recent frameworks extend portability to cross-family and cross-version transfer through algebraic subspace alignment, with or without lightweight fine-tuning (Farhadzadeh et al., 2025; Xia et al., 2025; Li et al., 2025), but none evaluate whether these techniques preserve control effectiveness for attribute-specific CTG modules specifically.

This paper addresses that gap, investigating whether attribute-specialised LoRA modules trained for sentiment and topic control can be ported across models that differ in architectural family, parameter count, or both.

2 Related Research

PEFT modules for CTG. Parameter-efficient fine-tuning (PEFT) methods, such as adapters, prefix tuning, and LoRA (Houlsby et al., 2019; Li and

Proceedings of the 19th International Natural Language Generation Conference: Long Papers, pages 1205–1218
October 17-21, 2026 ©2026 Association for Computational Linguistics

Liang, 2021; Hu et al., 2022), have been adopted as a control implementation strategy for CTG, injecting attribute-specific behaviour at a fraction of the cost of full fine-tuning. Prior CTG (Gu et al., 2022) conditions generation on learned attribute prefixes; GeDi (Krause et al., 2021) fine-tunes class-conditional language models whose contrastive predictions steer a separate base generator; DisCup (Zhang and Song, 2022) combines prompt-tuning with a control-attribute discriminator to optimise control prefixes for a fixed conditional LM. In each case, however, the tuned component is trained end-to-end for a specific attribute-value combination rather than isolated as an independently reusable module, so changing the target attribute requires retraining rather than swapping a single self-contained adapter.

More broadly, work on PEFT composability has shown that modules can be trained as standalone, swappable units combined at inference without joint retraining, as in AdapterFusion (Pfeiffer et al., 2021) and LoraHub (Huang et al., 2024), but this has not, to our knowledge, been applied to attribute-specific CTG control.

Portability of PEFT modules. PEFT techniques produce structurally modular parameter sets that interact with their host model via dedicated interfaces (Sabry and Belz, 2023), raising the question of whether this structural modularity also implies *functional* modularity: whether task-specific knowledge encoded in a module is self-contained enough to transfer to a different host without retraining. Sabry and Belz (2024) provide the foundational evidence for this, showing across 1,440 runs that PEFT modules ported between same-family host models substantially outperform both from-scratch training and randomly initialised baselines, though performance varies with the structural compatibility of the source-target pair. Cross-architecture transfer, however, is left open.

Subsequent work has targeted this gap along two lines. Training-free approaches transfer LoRA parameters algebraically by exploiting the geometric structure of weight spaces: LoRA-X (Farhadzadeh et al., 2025) constrains the adapter to the source model's column-row subspace and transfers it to target layers with sufficient subspace similarity. Cross-LoRA (Xia et al., 2025) aligns source and target subspaces via rank-truncated SVD before projecting the update into the target parameter space, demonstrating generalisation across heterogeneous families such as Qwen, LLaMA, and Gemma.

Where light retraining is acceptable, LoRASuite (Li et al., 2025) proposes vocabulary, hidden-dimension, layer-depth, and attention-head mismatches through a CKA-based layer mapping and Hungarian-matched head alignment, followed by brief fine-tuning to correct numerical instabilities, whereas Trans-LoRA (Wang et al., 2024) instead regenerates training data synthetically via the target model, filters it with a discriminator, and retrains a new LoRA from scratch, extending transfer across PEFT types (e.g. LoRA to DoRA).

Taken together, the above work establishes that knowledge transfer via LoRA modules is achievable across a spectrum of scenarios, from same-family transfers to cross-architecture settings, and from entirely training-free methods to those requiring lightweight adaptation. However, existing studies evaluate portability on general NLP tasks rather than on specific applied tasks such as attribute-specific CTG modules, a gap this paper addresses.

3 Study Overview

We assess module portability across three settings of increasing difficulty, defined by the relationship between source and target models:

1. **Same family, different size**: LLaMA 3.2 3B $\leftrightarrow$ LLaMA 3.1 8B, Qwen3 4B $\leftrightarrow$ Qwen3 8B;
2. **Different family, comparable size**: LLaMA 3.2 3B $\leftrightarrow$ Qwen3 4B, LLaMA 3.1 8B $\leftrightarrow$ Qwen3 8B; and
3. **Different family, different size**: LLaMA 3.2 3B $\leftrightarrow$ Qwen3 8B, LLaMA 3.1 8B $\leftrightarrow$ Qwen3 4B.

These settings successively add vocabulary, tokenisation, and attention-configuration mismatches on top of a baseline capacity gap. In each setting, each model pair is tested for porting in both directions.

For each of the four host models, namely LLaMA 3.2 3B, LLaMA 3.1 8B, Qwen3 4B, and Qwen3 8B (AI@Meta, 2024; Team, 2025), a single-attribute LoRA module is trained on the sentiment (Yelp) or topic (AGNews) control dataset and ported to all other hosts using the framework described in Section 6, yielding a full source-target transfer matrix. In each case, we evaluate both a ***zero-shot*** regime, where the ported module is used as-is, and a ***few-step*** regime, where a small amount of target-side fine-tuning follows transfer, testing whether portability is a viable alternative to retraining from scratch.

4 Task Construal

We construe attribute-controlled text generation as a prompted generation task, as this allows for a consistent and comparable evaluation across all four host models without requiring model-specific input formats or instruction-tuning. Each input consists of a leading text fragment tagged with the target attribute value(s), and the expected output is the text generated by the model in response, up to the first end-of-output tag or a specified maximum length, whichever comes first.

Leading fragments are constructed by taking the first n words of a given text from the in-domain datasets, $n = [0..5]$. Input/output pairs are then formed by splitting the tagged text after n words. For example, a training instance for sentiment-controlled generation takes the following form:

```
[SENTIMENT] Positive [\SENTIMENT] [ANS] The sushi is
great!  Not expensive & good quality.  My favorite rolles are the Vegas, Dragon, and the baked scallops :) [\ANS]
```

where the attribute tag and its value precede the answer tag `[ANS]`, which marks the beginning of the generated text.

5 PEFT Module Training

We train LoRA with Supervised Fine-Tuning (SFT) using the task construal described in Section 4. In vanilla single-module LoRA, a PEFT block consisting of a down-projection, and up-projection is attached in parallel to every key, query, value and feed-forward layer in every transformer block.

Adopting the notation from Asadi et al. (2024), LoRA learns a small set of task-specific parameters $\Delta\Theta$ in conjunction with the frozen pre-trained model Θ_0. $\Delta\Theta$ (the LoRA module) consists of trainable low-rank decomposition matrices added as adapter modules to the query, key, value and feed-forward weight matrices in the frozen model Θ_0. In other words, LoRA adds task-specific parameters $\mathbf{\Delta W}$ to pre-trained frozen weight matrices $\mathbf{W}_0 \in \mathbb{R}^{d\times c}$ in Θ_0:

$$\hat{\mathbf{W}} = \mathbf{W_0} + \frac{\alpha}{r}\mathbf{\Delta W}$$

where $\mathbf{\Delta W} = \mathbf{AB^T}$ is the LoRA block, with $\mathbf{A} \in \mathbb{R}^{d\times r}$ the down projection, $\mathbf{B}^T \in \mathbb{R}^{r\times d}$ the up projection, α a scaling factor, and r the rank.

$\mathbf{A}$ is initialised with the Kaiming uniform distribution, and $\mathbf{B}$ with all-zero initialisation. $\Delta\Theta$ is optimised with a standard conditional language modelling objective with CE loss computed over $\Theta_0 + \Delta\Theta$ (i.e. the frozen model with LoRA blocks plugged in). For a given input x, the output h resulting from adapting the frozen parameters $\mathbf{W_0}$ with a LoRA block is:

$$h = \mathbf{W_0}\, x + \frac{\alpha}{r}\mathbf{\Delta W} x$$

6 Portability Framework

We adapt LoRASuite (Li et al., 2025), originally designed for same-family version upgrades (e.g. LLaMA 2 to LLaMA 3), to the cross-family setting of this paper. The framework transfers LoRA weights algebraically through four steps. **(1) Embedding transformation:** a linear map $\mathbf{W}_h = \mathbf{E}^+_{\text{src}}\mathbf{E}_{\text{tgt}}$ is fitted between the source and target embedding matrices (restricted to shared vocabulary for cross-family pairs) to reconcile hidden-size and vocabulary mismatches; an analogous map $\mathbf{W}_i$ handles differing MLP intermediate dimensions. **(2) Layer mapping:** Centred Kernel Alignment (Kornblith et al., 2019) similarity between all source-target layer pairs, computed over a 256-text WikiText-2 calibration sample, is fed to a dynamic-programming alignment that maximises total similarity subject to a maximum offset ($\Delta_{\text{layer}} = 2$ same-family, 4 cross-family). **(3) Head mapping:** within each mapped layer pair, attention heads are matched via the Hungarian algorithm (Munkres, 1957) on the cosine similarity of their $\mathbf{W}^h_{QK} = \mathbf{W}^h_Q(\mathbf{W}^h_K)^\top$ and $\mathbf{W}^h_{VO} = \mathbf{W}^h_V(\mathbf{W}^h_O)^\top$ interaction matrices (projected through $\mathbf{W}_h$ when hidden sizes differ). **(4) Weight transformation:** for each mapped layer/head pair, the LoRA update is transformed as

$$(\Delta\mathbf{W}^{h_{\text{tgt}}})_{\text{tgt}} = \mathbf{W}_h^\top(\Delta\mathbf{W}^{h_{\text{src}}})_{\text{src}}(\mathbf{W}^{h_{\text{src}}})^\top_{\text{src}}\mathbf{W}_h(\mathbf{W}^{h_{\text{tgt}}})_{\text{tgt}},$$

incorporating the pretrained weights of both models for numerical stability, after which a single SVD recovers low-rank factors $\mathbf{A}_{\text{tgt}}, \mathbf{B}_{\text{tgt}}$ that attach directly to the frozen target model. This process requires no gradients or access to the original training data. In the few-step regime, the same algebraic transfer is followed by brief fine-tuning on the target model. For details see Appendix A.

7 Experimental Setup

The portability experiments use a dedicated set of single-attribute LoRA modules. We train on Yelp (Zhang et al., 2015) for sentiment control and AG News (Zhang et al., 2015) for topic control. One module per attribute is trained on each of the

four host models: LLaMA 3.2 3B, LLaMA 3.1 8B, Qwen3 4B, and Qwen3 8B.

To test the trained modules, we use two short-text datasets for out-of-domain evaluation only: PPLM Prompts (Dathathri et al., 2019) and STS benchmark test set (Cer et al., 2017). The latter (*STS* and *STS proc* in tables) provides single-sentence image captions from which we derive prompts in two ways: (i) using the whole sentence as a prompt (*STS*), and (ii) using the first n words, $n = 0..5$ (*STS proc*). Neither dataset comes with attribute labels; we therefore generate and evaluate one text per control attribute label for each prompt.

For hyperparameters see Appendix B.

Evaluation. We measure ***Diversity*** with Distinct-n (Li et al., 2015) which is the proportion of unique n-grams in generated texts. The aggregate Distinct-n score is the mean of item-level scores.

We assess ***Fluency*** with average Syntactic Log-Odds Ratio (SLOR) (Kann et al., 2018) obtained with GPT-2XL and BLOOM 1B7, in preference over perplexity which may not effectively capture fluency for low-frequency items. SLOR calculates the log-probability of a sentence normalised by unigram log-probability and length.

To assess ***Control Effectiveness*** (CE), we measure the mean percentage of texts identified by a set of classifiers as possessing the controlled attribute value. I.e. we first compute the percentage of cases for each classifier where the detected attribute value matches the input control value; the final CE score is then the average of the three classifiers' individual percentages. We use DistilBERT and T5 fine-tuned on SST-2, and DeBERTa fine-tuned on Yelp, for sentiment control; and DistilBERT, BERT, and DeBERTa fine-tuned on AG-News, for topic control (Appendix B.1).

Portability conditions. We report results for the following settings:

- **Raw model**: The pretrained model with no LoRA module attached and no fine-tuning of any kind, serving as a lower-bound baseline.
- **Identity**: The trained module attached to the model it was trained in, with no porting, serving as an upper-bound reference point.
- Module porting: The module is ported to the target model using the algebraic transfer described in Section 6. We further distinguish:
  - **Zero-shot porting**: The ported module is used directly for inference with no additional training.
  - **Few-step porting**: Following algebraic transfer, a small number of training steps are performed, using 10% of the original training data: 20 steps for sentiment control and 100 steps for topic control.

All experiments are run three times with different random seeds (4613, 7097, and 89723), and results are reported as averages across the three runs.

Code and results are available on GitHub.[1]

8 Results

This section presents the results for the experimental setup described above (Section 7), covering sentiment and topic control across the four host models and four portability conditions (Section 7). In the results tables below, the identity setting is displayed using the same arrow notation as the ported results (e.g. LLaMA 3.2 3B → LLaMA 3.2 3B) for consistency, but represents simply the fine-tuned module attached to the model it was originally trained in, with no porting involved. This serves as the upper-bound reference for each setting.

8.1 Sentiment control

Tables 1 and 2 present the portability results for sentiment control under zero-shot and few-step porting, respectively, across all four target models and all transfer conditions. Identity configurations achieve 84.1–85.2% control efficiency (CE) across the four target models, considerably above raw-model baselines of 56.0–68.4%. This confirms that the trained modules encode strong task-specific knowledge, and establishes the topline against which ported results are interpreted.

Zero-shot porting. Results stay close to the raw-model baseline in every transfer condition (Table 1). Within the LLaMA family, LLaMA 3.1 8B → LLaMA 3.2 3B reaches 60.8% CE and LLaMA 3.2 3B → LLaMA 3.1 8B reaches 57.4%, against baselines of 60.6% and 56.0% respectively, essentially no improvement over the model on its own.

Cross-family porting to LLaMA models score 50.0–60.7% CE, again near baseline, with Qwen3 8B → LLaMA 3.1 8B (50.0%) falling *below* its 56.0% baseline, indicating that it is possible for

[1] https://github.com/michelalorandi/portable_lora

Setting	Control Effectiveness ↑				Distinct-n ↑			SLOR ↑	Params
	Avg all	PPLM	STS	STS proc	dist-1	dist-2	dist-3		
SENTIMENT CONTROL ZERO-STEP PORTING									
Target: LLaMA 3.2 3B									
LLaMA 3.2 3B *(no module)*	60.6	*71.0*	*58.7*	59.0	0.306	0.720	0.815	*9.77*	—
LLaMA 3.2 3B → LLaMA 3.2 3B	**85.2**	**90.0**	**80.1**	**88.7**	0.263	**0.760**	**0.892**	**10.02**	1.0×
LLaMA 3.1 8B → LLaMA 3.2 3B	*60.8*	68.4	57.7	61.3	0.312	0.727	0.822	9.73	0.4×
Qwen3 4B → LLaMA 3.2 3B	60.7	63.7	58.4	*61.9*	*0.317*	0.729	0.819	9.70	0.8×
Qwen3 8B → LLaMA 3.2 3B	60.3	66.0	58.4	60.2	**0.317**	*0.741*	*0.830*	9.75	0.4×
Target: LLaMA 3.1 8B									
LLaMA 3.1 8B *(no module)*	56.0	62.9	53.2	56.5	*0.326*	**0.764**	*0.867*	9.65	—
LLaMA 3.1 8B → LLaMA 3.1 8B	**84.1**	**90.6**	**78.4**	**87.6**	0.267	0.761	**0.886**	**9.90**	1.0×
LLaMA 3.2 3B → LLaMA 3.1 8B	*57.4*	*63.0*	53.9	*59.0*	0.323	0.744	0.833	9.68	2.7×
Qwen3 8B → LLaMA 3.1 8B	50.0	48.7	50.6	49.9	0.273	0.707	0.812	9.46	1.0×
Qwen3 4B → LLaMA 3.1 8B	57.1	59.4	*55.0*	58.4	**0.329**	*0.761*	0.852	*9.70*	2.0×
Target: Qwen3 4B									
Qwen3 4B *(no module)*	68.4	68.1	65.3	*71.7*	0.078	*0.438*	*0.656*	*9.14*	—
Qwen3 4B → Qwen3 4B	**84.7**	**90.6**	**81.0**	**86.4**	**0.173**	**0.661**	**0.844**	**9.84**	1.0×
Qwen3 8B → Qwen3 4B	68.4	*70.5*	*67.2*	68.9	*0.079*	0.428	0.643	9.11	0.5×
LLaMA 3.2 3B → Qwen3 4B	*68.7*	66.2	66.9	71.3	0.076	0.419	0.634	9.10	1.3×
LLaMA 3.1 8B → Qwen3 4B	68.4	68.4	66.8	70.0	0.079	0.423	0.641	9.06	0.5×
Target: Qwen3 8B									
Qwen3 8B *(no module)*	58.5	*66.2*	56.5	57.8	0.111	0.554	0.739	9.23	—
Qwen3 8B → Qwen3 8B	**85.1**	**93.5**	**80.8**	**86.4**	**0.214**	**0.699**	**0.856**	**9.73**	1.0×
Qwen3 4B → Qwen3 8B	57.9	62.4	58.2	56.1	0.111	0.561	0.736	9.30	2.0×
LLaMA 3.1 8B → Qwen3 8B	56.8	62.2	55.6	56.2	*0.120*	*0.584*	*0.752*	*9.49*	1.0×
LLaMA 3.2 3B → Qwen3 8B	*60.9*	65.9	*58.7*	*61.2*	0.114	0.563	0.738	9.32	2.7×

Table 1: Portability results with 0 post-porting learning steps for sentiment control, grouped by target model. Each block shows the raw base model (no module), the identity upper bound, and all ported configurations. CE values are percentages averaged over seeds; Distinct-n and SLOR are averaged over test sets and seeds. Params = size of the target model relative to the source model. **Bold** = best; *Italic* = second best within each block.

transfer to degrade performance rather than merely failing to help. Porting to Qwen models also doesn't achieve much of an improvement: porting to Qwen3 4B yields 68.4–68.7% CE against a 68.4% baseline, to Qwen3 8B, 56.8–60.9% against a 58.5% baseline. The best case (LLaMA 3.2 3B → Qwen3 8B, 60.9%) still falls far short of the 85.1% identity score. Diversity and fluency metrics track the raw-model baselines throughout, with no consistent gain attributable to the ported module. Overall, zero-shot algebraic transfer does not reliably inject sentiment control capabilities into the new host in any of the transfer conditions tested.

Few-step porting. In contrast, adding lightweight (10% training data, 20 learning steps) fine-tuning (LFT) after transfer not only performs substantially better than the target model by itself, it even closes much of the gap to the identity setting (Table 2), with average CE ('Avg all' column) rising to 52.4–81.3%. For LLaMA 3.2 3B as the target model, all source models reach 76.2–78.7% CE (best: Qwen3 4B → LLaMA 3.2 3B, 78.7%), around 6.5 points short of the 85.2% identity score. For LLaMA 3.1 8B targets, most sources reach 80.7–81.3%, within 3–4 points of the 84.1% identity score, except Qwen3 8B → LLaMA 3.1 8B, which remains at 52.4%, below the raw baseline and unresolved by fine-tuning, confirming this as a persistent incompatible pair.

For Qwen targets, few-step porting reaches 72.4–76.0% CE for Qwen3 8B and 75.7–78.1% for Qwen3 4B, gains of 12–16 points over zero-shot but still 7–12 points below identity scores (85.1% and 84.7%); the best sources are LLaMA 3.1 8B in both cases (76.0% and 78.1%). Distinct-n scores fall relative to the raw baselines and move closer to identity levels, while SLOR remains stable.

8.2 Topic control

Tables 3 and 4 show the corresponding results for topic control under zero-shot and few-step porting, respectively. The identity configurations achieve CE scores between 56.2% and 75.2% across the four target models, compared to raw model baselines of 26.2%–35.6%, establishing the upper and lower bounds for the ported results, respectively. The gap between these bounds is larger than that ob-

Sentiment control few-step porting									
Setting	**LFT**	**Control Effectiveness ↑**				**Distinct-n ↑**			**SLOR ↑**
		Avg all	**PPLM**	**STS**	**STS proc**	**dist-1**	**dist-2**	**dist-3**	
Target: LLaMA 3.2 3B									
LLaMA 3.2 3B *(no module)*	×	60.6	71.0	58.7	59.0	**0.306**	*0.720*	*0.815*	9.77
LLaMA 3.2 3B → LLaMA 3.2 3B	×	**85.2**	**90.0**	**80.1**	**88.7**	*0.263*	**0.760**	**0.892**	**10.02**
LLaMA 3.1 8B → LLaMA 3.2 3B	✓	76.2	80.3	72.9	78.2	0.187	0.642	0.812	*9.84*
Qwen3 4B → LLaMA 3.2 3B	✓	*78.7*	80.5	*74.9*	*81.9*	0.184	0.636	0.810	9.83
Qwen3 8B → LLaMA 3.2 3B	✓	77.4	*82.7*	73.9	78.9	0.186	0.640	0.812	9.82
Target: LLaMA 3.1 8B									
LLaMA 3.1 8B *(no module)*	×	56.0	62.9	53.2	56.5	**0.326**	**0.764**	*0.867*	9.65
LLaMA 3.1 8B → LLaMA 3.1 8B	×	**84.1**	**90.6**	**78.4**	**87.6**	*0.267*	*0.761*	**0.886**	**9.90**
LLaMA 3.2 3B → LLaMA 3.1 8B	✓	80.7	88.1	73.7	*85.1*	0.188	0.643	0.814	*9.81*
Qwen3 8B → LLaMA 3.1 8B	✓	52.4	53.2	50.1	54.3	0.149	0.524	0.706	9.51
Qwen3 4B → LLaMA 3.1 8B	✓	*81.3*	*89.4*	*75.4*	84.3	0.183	0.632	0.805	9.81
Target: Qwen3 4B									
Qwen3 4B *(no module)*	×	68.4	68.1	65.3	71.7	0.078	0.438	0.656	9.14
Qwen3 4B → Qwen3 4B	×	**84.7**	**90.6**	**81.0**	**86.4**	**0.173**	**0.661**	**0.844**	9.84
Qwen3 8B → Qwen3 4B	✓	75.7	81.4	72.2	77.3	*0.172*	*0.606*	0.782	9.89
LLaMA 3.2 3B → Qwen3 4B	✓	77.5	83.8	*73.3*	79.6	0.164	0.604	*0.786*	**9.95**
LLaMA 3.1 8B → Qwen3 4B	✓	*78.1*	*86.0*	73.1	*80.2*	0.163	0.601	0.783	*9.92*
Target: Qwen3 8B									
Qwen3 8B *(no module)*	×	58.5	66.2	56.5	57.8	0.111	0.554	0.739	9.23
Qwen3 8B → Qwen3 8B	×	**85.1**	**93.5**	**80.8**	**86.4**	**0.214**	**0.699**	**0.856**	9.73
Qwen3 4B → Qwen3 8B	✓	73.6	81.7	69.4	75.1	*0.191*	*0.652*	*0.818*	*9.85*
LLaMA 3.1 8B → Qwen3 8B	✓	*76.0*	*84.1*	69.4	*79.6*	0.176	0.631	0.812	9.79
LLaMA 3.2 3B → Qwen3 8B	✓	72.4	83.3	*69.8*	71.2	0.184	0.643	0.816	**9.88**

Table 2: Portability results with 20 post-porting learning steps (10% of training data) for sentiment control grouped by target model. Each block shows the raw base model (no module), the identity upper bound, and all source models ported into the target. All values are averaged over 3 seeds. CE values are percentages; Distinct-n and SLOR are weighted averages across test sets and seeds. **Bold** = best within each transfer condition, *Italic* = second best. LFT = post-porting Lightweight Fine-Tuning.

served for sentiment control, reflecting the greater difficulty of topic control.

Zero-shot porting. CE scores sit at or below baseline throughout (Table 3), ranging from 25.1% to 34.5% against baselines of 26.2–35.6%. Two pairs fall *below* baseline with a marked loss of fluency: Qwen3 8B → LLaMA 3.1 8B (25.1% CE, SLOR 5.16, against a 35.6% baseline) and LLaMA 3.1 8B → Qwen3 8B (25.4% CE, SLOR 7.39, against a 26.2% baseline), indicating degradation rather than mere failure to transfer. Transfers into the Qwen models score 25.4–26.3% CE for Qwen3 8B and 26.8–27.2% for Qwen3 4B, indistinguishable from their 26.2% and 26.7% baselines; transfers into the LLaMA models score 31.7–33.5% for LLaMA 3.2 3B and 25.1–34.5% for LLaMA 3.1 8B, likewise tracking baseline. Overall, as was observed for sentiment control, zero-shot porting injects no meaningful topic control capability into the target host model in any setting tested.

Few-step porting. Here too, few-step porting with lightweight post-porting tuning (10% of the training data, 100 learning steps) recovers much more ground than the zero-shot condition (Table 4), average CE ranging from 26.3% to 59.2%, mostly clustering between 49.6% and 59.2%. For LLaMA 3.2 3B targets, all sources reach 54.9–57.1% CE (best: LLaMA 3.1 8B → LLaMA 3.2 3B, 57.1%), about 17.5 points below the 74.6% identity score. For LLaMA 3.1 8B targets, most sources reach 57.5–59.2%, 10–12 points below the 69.8% identity score, except Qwen3 8B → LLaMA 3.1 8B, which remains low at 26.3% CE with a SLOR of 4.15, the same failure case observed under zero-shot porting, again unresolved by fine-tuning.

For the Qwen targets, few-step porting reaches 49.6–50.6% CE for Qwen3 4B (17 points below its 67.0% identity score) and 51.3–54.4% for Qwen3 8B, with the best case, LLaMA 3.1 8B → Qwen3 8B at 54.4%, coming within roughly 1 point of the 55.6% identity score, the closest recovery observed across the topic control experiments. As with sentiment control, Distinct-n falls and moves closer to identity levels while SLOR stays broadly stable, except for the Qwen3 8B → LLaMA 3.1 8B

TOPIC CONTROL ZERO-STEP PORTING									
Setting	**Control Effectiveness ↑**				**Distinct-n ↑**			**SLOR ↑**	**Params**
	Avg all	**PPLM**	**STS**	**STS proc**	**dist-1**	**dist-2**	**dist-3**		
Target: LLaMA 3.2 3B									
LLaMA 3.2 3B *(no module)*	33.2	36.9	*30.7*	34.4	*0.314*	*0.750*	*0.840*	9.38	—
LLaMA 3.2 3B → LLaMA 3.2 3B	**74.6**	**85.8**	**63.8**	**81.5**	0.263	0.629	0.723	9.20	1.0×
LLaMA 3.1 8B → LLaMA 3.2 3B	31.8	36.2	29.4	32.7	0.296	0.731	0.823	*9.45*	0.4×
Qwen3 4B → LLaMA 3.2 3B	*33.5*	35.2	30.4	*36.1*	**0.320**	**0.764**	**0.853**	**9.49**	0.8×
Qwen3 8B → LLaMA 3.2 3B	31.7	*37.8*	28.4	32.8	0.306	0.745	0.839	9.39	0.4×
Target: LLaMA 3.1 8B									
LLaMA 3.1 8B *(no module)*	*35.6*	38.3	30.8	*39.4*	*0.304*	**0.748**	**0.843**	*9.35*	—
LLaMA 3.1 8B → LLaMA 3.1 8B	**69.8**	**85.5**	**56.9**	**77.2**	0.267	0.626	0.718	9.14	1.0×
LLaMA 3.2 3B → LLaMA 3.1 8B	34.1	*41.0*	*31.0*	34.8	0.303	0.731	0.822	9.31	2.7×
Qwen3 8B → LLaMA 3.1 8B	25.1	25.6	25.1	25.0	0.054	0.279	0.495	5.16	1.0×
Qwen3 4B → LLaMA 3.1 8B	34.5	40.3	30.9	35.9	**0.308**	*0.742*	*0.835*	**9.43**	2.0×
Target: Qwen3 4B									
Qwen3 4B *(no module)*	26.7	28.6	*25.7*	27.0	0.095	0.506	0.733	9.17	—
Qwen3 4B → Qwen3 4B	**67.0**	**84.8**	**52.6**	**75.2**	**0.275**	**0.642**	*0.737*	9.07	1.0×
Qwen3 8B → Qwen3 4B	*27.2*	29.1	25.4	26.6	*0.123*	*0.555*	**0.768**	**9.26**	0.5×
LLaMA 3.2 3B → Qwen3 4B	26.8	*29.4*	25.3	*27.4*	0.097	0.496	0.722	*9.22*	1.3×
LLaMA 3.1 8B → Qwen3 4B	26.8	29.2	25.4	27.3	0.093	0.491	0.720	9.18	0.5×
Target: Qwen3 8B									
Qwen3 8B *(no module)*	26.2	27.9	25.2	26.6	0.128	*0.634*	**0.798**	*9.39*	—
Qwen3 8B → Qwen3 8B	**55.6**	**77.1**	**38.7**	**65.1**	**0.297**	**0.696**	*0.791*	9.17	1.0×
Qwen3 4B → Qwen3 8B	*26.3*	*27.9*	25.3	*26.7*	0.128	0.623	0.786	9.32	2.0×
LLaMA 3.1 8B → Qwen3 8B	25.4	25.6	25.3	25.5	0.058	0.389	0.607	7.39	1.0×
LLaMA 3.2 3B → Qwen3 8B	26.2	27.3	*25.3*	26.7	*0.132*	0.625	0.785	**9.51**	2.7×

Table 3: Portability results with 0 post-porting learning steps for topic control grouped by target model. Each block shows the raw base model (no module), the identity upper bound, and all source models ported into the target. All values are averaged over 3 seeds. CE values are percentages; Distinct-n and SLOR are weighted averages across test sets and seeds. **Bold** = best within each transfer condition, *Italic* = second best.

outlier; overall, few-step porting yields meaningful but incomplete recovery, with this pair remaining a persistent failure case under both porting regimes.

9 Discussion

Our aim has been to investigate if attribute-specialised LoRA modules trained on one host model can be ported to a different target model without retraining from scratch, across three transfer conditions of increasing difficulty: same-family, different size; cross-family, comparable size; and cross-family, different size, for sentiment and topic control, with (few-step) and without (zero-shot) lightweight finetuning applied post porting.

Why zero-shot transfer falls short. Ported modules consistently perform at or near the raw model baseline under zero-shot porting, in all transfer settings and both control attributes, indicating that algebraic transfer alone does not reliably preserve task-specific adaptation in the settings tested. This is not a matter of same-family transfers succeeding while cross-family transfers fail, as this is the same in both cases. This suggests that the layer and head-mapping steps of the framework, while structurally sound, do not on their own carry enough of the fine-tuned signal across to the target parameter space without some corrective adaptation. Strikingly, two transfer settings, Qwen3 8B → LLaMA 3.1 8B for sentiment and for topic control, fall *below* the raw model baseline, in the topic control case additionally showing a marked drop in fluency (SLOR). This points to something beyond a simple failure to transfer: the algebraic transformation appears to actively perturb the target model's behaviour, rather than leaving it merely unchanged. A plausible explanation is that the head and layer-mapping steps, calibrated on general-domain text, do not adequately capture the specific representational geometry that would make this source-target pair compatible, though confirming this would require further probing of the intermediate CKA and head-similarity scores for this model pair.

What few-step porting recovers, and what it does not. Lightweight post-porting fine-tuning changes the picture, succeeding in recovering a good proportion of the gap left by zero-shot trans-

TOPIC CONTROL FEW-STEP PORTING									
Setting	**LFT**	**Control Effectiveness ↑**				**Distinct-n ↑**			**SLOR ↑**
		Avg all	**PPLM**	**STS**	**STS proc**	**dist-1**	**dist-2**	**dist-3**	
Target: LLaMA 3.2 3B									
LLaMA 3.2 3B *(no module)*	×	33.2	36.9	30.7	34.4	**0.314**	**0.750**	**0.840**	9.38
LLaMA 3.2 3B → LLaMA 3.2 3B	×	**74.6**	**85.8**	**63.8**	**81.5**	0.263	0.629	0.723	9.20
LLaMA 3.1 8B → LLaMA 3.2 3B	✓	*57.1*	*75.6*	*41.6*	*66.1*	0.270	*0.697*	*0.815*	*9.49*
Qwen3 4B → LLaMA 3.2 3B	✓	55.3	75.2	39.5	64.2	0.264	0.683	0.799	9.42
Qwen3 8B → LLaMA 3.2 3B	✓	54.9	73.6	40.2	63.0	*0.271*	0.695	0.811	**9.50**
Target: LLaMA 3.1 8B									
LLaMA 3.1 8B *(no module)*	×	35.6	38.3	30.8	39.4	**0.304**	**0.748**	**0.843**	**9.35**
LLaMA 3.1 8B → LLaMA 3.1 8B	×	**69.8**	**85.5**	**56.9**	**77.2**	0.267	0.626	0.718	9.14
LLaMA 3.2 3B → LLaMA 3.1 8B	✓	57.5	76.3	41.6	66.8	0.264	0.648	0.754	9.20
Qwen3 8B → LLaMA 3.1 8B	✓	26.3	26.2	26.0	26.7	0.143	0.323	0.399	4.15
Qwen3 4B → LLaMA 3.1 8B	✓	*59.2*	*79.5*	*43.7*	*67.5*	*0.276*	*0.673*	*0.780*	*9.35*
Target: Qwen3 4B									
Qwen3 4B *(no module)*	×	26.7	28.6	25.7	27.0	0.095	0.506	0.733	9.17
Qwen3 4B → Qwen3 4B	×	**67.0**	**84.8**	**52.6**	**75.2**	**0.275**	0.642	0.737	9.07
Qwen3 8B → Qwen3 4B	✓	*50.6*	67.8	*38.4*	*56.9*	0.258	0.682	0.812	9.56
LLaMA 3.2 3B → Qwen3 4B	✓	49.6	*68.5*	37.0	55.5	*0.264*	**0.695**	**0.824**	**9.61**
LLaMA 3.1 8B → Qwen3 4B	✓	50.0	66.5	37.6	56.6	0.264	*0.695*	*0.824*	*9.56*
Target: Qwen3 8B									
Qwen3 8B *(no module)*	×	26.2	27.9	25.2	26.6	0.128	0.634	0.798	9.39
Qwen3 8B → Qwen3 8B	×	**55.6**	**77.1**	38.7	**65.1**	**0.297**	**0.696**	0.791	9.17
Qwen3 4B → Qwen3 8B	✓	51.3	66.3	38.1	59.4	0.254	0.654	0.769	9.40
LLaMA 3.1 8B → Qwen3 8B	✓	*54.4*	*73.7*	**41.1**	61.1	0.254	0.683	**0.814**	*9.49*
LLaMA 3.2 3B → Qwen3 8B	✓	53.6	73.3	*39.0*	*61.4*	*0.268*	*0.693*	*0.813*	**9.54**

Table 4: Portability results with 100 post-porting learning steps (10% of training data) for topic control grouped by target model. Each block shows the raw base model (no module), the identity upper bound, and all source models ported into the target. All values are averaged over 3 seeds. CE values are percentages; Distinct-n and SLOR are weighted averages across test sets and seeds. **Bold** = best within each transfer condition, *Italics* = second best. LFT = post-porting Lightweight Fine-Tuning.

fer. This suggests that the algebraic transfer step, even when insufficient on its own, provides a useful initialisation that a few gradient steps can refine, moreover with a fraction of the training data, into a genuinely useful task-adapted module, at considerably less compute than training from scratch. Recovery is not, however, uniform: it is more complete for sentiment than for topic control, consistent with topic control being the harder task overall. Furthermore, the Qwen3 8B → LLaMA 3.1 8B pair is an underpeforming outlier under few-step porting as well, for both sentiment and topic control. That a short adaptation phase closes the gap in most pairings but leaves this one essentially untouched suggests the underlying issue is not simply a matter of insufficient adaptation steps, but a more fundamental incompatibility between this specific source and target representation space that the current framework does not detect or correct for.

10 Conclusion

Taken together, our results provide initial evidence of the feasibility of cross-model portability for attribute-specialised PEFT modules. Zero-shot porting, in its current form, does not achieve the goal of training-free module reuse across the model pairs tested here, and for one model pair appears to degrade performance relative to the untuned model. Few-step porting is considerably more promising, recovering a substantial share of task-specific performance with only a small target-side training budget. This makes it a practically appealing configuration when a source module is already available, though we note that our experiments do not include a baseline in which a fresh LoRA module is trained on the target model from scratch under the same data and step budget; we can therefore not yet establish how much of this recovery is attributable to the algebraic transfer step itself, as opposed to the effect of a small amount of target-side training regardless of initialisation. A residual gap to the identity configuration remains in most settings, however, and the persistent Qwen3 8B → LLaMA 3.1 8B failure case indicates that algebraic alignment alone cannot guarantee compatibility for every source-target pair.

Importantly, our results are consistent with structurally modular PEFT modules retaining some functional modularity: removing a module and inserting it into a new host model, together with a minimal amount of adaptation tuning to the new host environment, can deliver much of the original task-specific performance. Whether this reflects genuine transfer of task-specific knowledge through the algebraic alignment step, or is instead better explained by the adaptation step alone, is a question our current experimental design cannot fully disentangle, and one we consider an important direction for follow-up work.

Ethical Considerations

Data. All datasets used in this work are publicly available and widely used in prior CTG research. Yelp reviews are user-generated but are distributed as an established public benchmark with no additional personal or sensitive information collected or inferred beyond what the original release contains; no attempt is made to identify or profile individual reviewers. No new data was collected from people as part of this study.

Models. All base models used (LLaMA 3.2 3B, LLaMA 3.1 8B, Qwen3 4B, and Qwen3 8B) are publicly released and used in accordance with their respective licences. No proprietary or closed models are used, supporting the reproducibility of this work.

Dual use. Controlled text generation, and by extension the portability of attribute-control modules studied here, carries a general dual-use risk: the same mechanisms that allow a module to reliably steer generation towards a positive sentiment or a specific topic could equally be used to produce persuasive, manipulative, or misleading content at scale, for instance for targeted disinformation. Our portability framework does not introduce new generative capabilities beyond what is already achievable by fine-tuning each base model directly; rather, it investigates whether an already-trained module can be reused across models without retraining. This lowers the operational cost of applying a given attribute-control module to additional host models, which is a benefit for legitimate reuse but could equally reduce the effort required to redeploy a harmful module across multiple models. We are not aware of any way to prevent this class of risk that is specific to portability rather than to attribute-controlled generation in general, and we note that the attributes studied here (sentiment and topic) are intentionally chosen as low-risk, well-established CTG benchmarks rather than as attributes associated with harmful content.

Reproducibility. Code and results are available on GitHub[2] to allow the community to verify, extend, and build on this work, including the failure case identified for the Qwen3 8B → LLaMA 3.1 8B transfer pair, which we believe merits further independent investigation.

Limitations

The portability experiments are subject to some limitations. First of all, the evaluation covers only two control attributes, sentiment and topic, using a single training dataset per attribute. Whether the observed portability patterns extend to other attributes, more complex generation tasks, or modules trained on different datasets remains to be established and is left to future work.

The portability framework is applied here to four models across two families, covering three transfer conditions of increasing difficulty. While this design enables a structured comparison across conditions, the number of model pairs tested is limited, and results may not generalise to other model families or to transfers involving larger architectural differences than those considered here.

The few-step porting condition uses a fixed proportion of the original training data (10%) and a fixed number of learning steps for each attribute. The sensitivity of the results to these choices is not explored, and different data scales or learning schedules may yield different outcomes.

Generation quality is assessed only via Distinct-n and SLOR, which capture lexical diversity and fluency but do not directly measure coherence, relevance, repetition beyond the n-gram level, semantic preservation of the input prompt, or the naturalness with which the target attribute is expressed. A more complete quality assessment, potentially including human evaluation, is left to future work.

The proposed framework combines four components (embedding transformation, CKA-based layer mapping, Hungarian head matching, and pretrained-weight-informed transformation followed by SVD reconstruction), none of which is

[2] https://github.com/michelalorandi/portable_lora

individually ablated here; we therefore cannot attribute the framework's zero-shot and few-step results to any specific component, nor rule out that a simpler transfer strategy (e.g. direct copying, proportional layer mapping, or random mapping) would perform comparably. Similarly, while Section 2 reviews LoRA-X, Cross-LoRA, and TransLoRA as alternative portability frameworks, our experiments evaluate only the adapted LoRASuite pipeline; a direct empirical comparison against these alternatives is left to future work.

While all experiments are repeated across three random seeds, we report only seed-averaged scores in the main text and do not provide standard deviations, confidence intervals, or significance tests. This is a limitation of the current reporting rather than of the underlying experiments; per-seed results are included in the released code and data for independent verification.

Given the limited scope of this study, two control attributes, four host models, and a single portability framework, evaluated with a narrow set of quality metrics and without component-level ablation, the findings of this work should be read as preliminary evidence rather than a general characterisation of cross-model portability. Future work could usefully investigate whether the observed failure case reflects a diagnosable property of the source-target pair (e.g. detectable from CKA or head-similarity scores ahead of time), whether alternative algebraic transfer strategies, or post-porting training with more data and/or more learning steps can close the residual gap left by few-step porting, and whether these findings generalise beyond the LLaMA-Qwen family pairings and sentiment/topic attributes studied here.

Our task construal relies on literal control tags (e.g. `[SENTIMENT]`, `[ANS]`) that are tokenised differently by LLaMA and Qwen's respective tokenisers. Consequently, some portion of the degraded cross-family transfer we observe may reflect a failure to transfer the learned control *interface* (i.e. how the tag sequence is used to trigger controlled generation) rather than a failure to transfer the underlying attribute knowledge itself. Disentangling these two sources of degradation, for instance by using a tag-free or embedding-level control signal shared across tokenisers, is left to future work.

A further limitation concerns the few-step porting comparison: we do not include a baseline in which a fresh LoRA module is initialised on the target model and trained with the same 10% data budget, step count, optimiser, and seeds as the few-step porting condition. Without this baseline, we cannot establish how much of the few-step porting result is attributable to the algebraic transfer step itself, as opposed to simply the effect of a small amount of target-side training regardless of initialisation. This is an important comparison for future work, and we note it as a limitation of the present study rather than a settled claim: our results should be read as showing that few-step porting *achieves* strong performance, not that porting-then-tuning outperforms tuning-from-scratch under matched budgets.

Acknowledgements

Michela Lorandi's work was conducted with the financial support of the Science Foundation Ireland Centre for Research Training in Digitally-Enhanced Reality (d-real) under Grant No. 18/CRT/6224. Both authors benefit from being members of the ADAPT SFI Research Centre at Dublin City University, funded by the Science Foundation Ireland under Grant Agreement No. 13/RC/2106_P2.

References

AI@Meta. 2024. Llama 3 model card.

Nader Asadi, Mahdi Beitollahi, Yasser H Khalil, Yinchuan Li, Guojun Zhang, and Xi Chen. 2024. Combining pre-trained lora modules improves few-shot adaptation of foundation models to new tasks. In *ICML 2024 Workshop on Foundation Models in the Wild*.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2019. Plug and play language models: A simple approach to controlled text generation. *arXiv preprint arXiv:1912.02164*.

Farzad Farhadzadeh, Debasmit Das, Shubhankar Borse, and Fatih Porikli. 2025. LoRA-x: Bridging foundation models with training-free cross-model adaptation. In *The Thirteenth International Conference on Learning Representations*.

Yuxuan Gu, Xiaocheng Feng, Sicheng Ma, Lingyuan Zhang, Heng Gong, and Bing Qin. 2022. A distri-

butional lens for multi-aspect controllable text generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1023–1043, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yuxuan Gu, Xiaocheng Feng, Sicheng Ma, Lingyuan Zhang, Heng Gong, Weihong Zhong, and Bing Qin. 2023. Controllable text generation via probability density estimation in the latent space. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12590–12616, Toronto, Canada. Association for Computational Linguistics.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2022. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Chengsong Huang, Qian Liu, Bill Yuchen Lin, Tianyu Pang, Chao Du, and Min Lin. 2024. Lorahub: Efficient cross-task generalization via dynamic loRA composition. In *First Conference on Language Modeling*.

Katharina Kann, Sascha Rothe, and Katja Filippova. 2018. Sentence-level fluency evaluation: References help, but can be spared! In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 313–323, Brussels, Belgium. Association for Computational Linguistics.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. 2019. Similarity of neural network representations revisited. In *International conference on machine learning*, pages 3519–3529. PMlR.

Ben Krause, Akhilesh Deepak Gotmare, Bryan McCann, Nitish Shirish Keskar, Shafiq Joty, Richard Socher, and Nazneen Fatema Rajani. 2021. GeDi: Generative discriminator guided sequence generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4929–4952, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2015. A diversity-promoting objective function for neural conversation models. *arXiv preprint arXiv:1510.03055*.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *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)*, pages 4582–4597, Online. Association for Computational Linguistics.

Yanan Li, Fanxu Meng, Muhan Zhang, Shiai Zhu, Shangguang Wang, and Mengwei Xu. 2025. LoRASuite: Efficient loRA adaptation across large language model upgrades. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

I Loshchilov. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

James Munkres. 1957. Algorithms for the assignment and transportation problems. *Journal of the society for industrial and applied mathematics*, 5(1):32–38.

Thao Nguyen, Maithra Raghu, and Simon Kornblith. 2021. Do wide and deep networks learn the same things? uncovering how neural network representations vary with width and depth. In *International Conference on Learning Representations*.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. 2021. AdapterFusion: Non-destructive task composition for transfer learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 487–503, Online. Association for Computational Linguistics.

Mohammed Sabry and Anya Belz. 2023. Peft-ref: A modular reference architecture and typology for parameter-efficient finetuning techniques. *arXiv preprint arXiv:2304.12410*.

Mohammed Sabry and Anya Belz. 2024. Assessing the portability of parameter matrices trained by parameter-efficient finetuning methods. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 1548–1556, St. Julian's, Malta. Association for Computational Linguistics.

Qwen Team. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Runqian Wang, Soumya Ghosh, David Cox, Diego Antognini, Aude Oliva, Rogerio Feris, and Leonid Karlinsky. 2024. Trans-LoRA: towards data-free transferable parameter efficient finetuning. *Advances in Neural Information Processing Systems*, 37:61217–61237.

Feifan Xia, Mingyang Liao, Yuyang Fang, Defang Li, Yantong Xie, Weikang Li, Yang Li, Deguo Xia, and Jizhou Huang. 2025. Cross-lora: A data-free lora transfer framework across heterogeneous llms. *arXiv preprint arXiv:2508.05232*.

Hanqing Zhang and Dawei Song. 2022. DisCup: Discriminator cooperative unlikelihood prompt-tuning for controllable text generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3392–3406, Abu

Dhabi, United Arab Emirates. Association for Computational Linguistics.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

A Portability Framework

The portability framework adapts LoRASuite (Li et al., 2025), a modular approach originally designed to transfer LoRA weights across successive versions of the same model family, and applies it to the attribute-controlled text generation setting described in this paper. While the original LoRASuite work focuses on same-family, same-architecture upgrades (e.g. LLaMA 2 to LLaMA 3), we adapt it, and apply it to, the three transfer conditions of increasing difficulty introduced in Section 3, which additionally include cross-family transfer between the LLaMA and Qwen model families. In the zero-shot porting regime, the framework operates without any retraining on the target model, transferring the LoRA weights algebraically through a pipeline of four sequential steps, shown in pseudo-code in Algorithm 1.

A.1 Step 1: Embedding Transformation

The first step addresses dimensional mismatches between the source and target model's hidden representation spaces, which arise when the two models differ in vocabulary size, hidden size, or both. Based on LoRASuite (Li et al., 2025), we build a linear mapping $\mathbf{W}_h \in \mathbb{R}^{d_{\text{src}} \times d_{\text{tgt}}}$ that converts representations from the source model's input space into the target model's input space. It is computed from the embedding weight matrices of both models as:

$$\mathbf{W}_h = \mathbf{E}_{\text{src}}^{+} \mathbf{E}_{\text{tgt}} \tag{1}$$

where $\mathbf{E}_{\text{src}}$ and $\mathbf{E}_{\text{tgt}}$ are the input embedding matrices of the source and target models, respectively, and $\mathbf{E}_{\text{src}}^{+}$ is the Moore-Penrose pseudoinverse of $\mathbf{E}_{\text{src}}$, obtained by solving a least-squares problem. Concretely, this finds the matrix $\mathbf{W}_h$ that makes $\mathbf{E}_{\text{src}}\mathbf{W}_h$ best approximate $\mathbf{E}_{\text{tgt}}$ in the least-squares sense, so that applying $\mathbf{W}_h$ maps source embeddings to the closest matching vectors in the target embedding space. When the two models use different vocabularies, as is the case for cross-family transfer between LLaMA and Qwen, only the shared token subset is used to estimate $\mathbf{W}_h$. If the hidden sizes are identical and the vocabulary fully overlaps, $\mathbf{W}_h$ reduces to the identity matrix and this step is skipped. For up- and down-projection layers whose intermediate dimensions differ, a separate transformation matrix $\mathbf{W}_i$ is derived from the projection weights of both models using the same least-squares approach.

A.2 Step 2: CKA-Based Layer Mapping

The second step addresses the structural mismatch arising when the source and target models have different numbers of transformer layers. Since different layers capture representations at different levels of abstraction, a simple sequential alignment (mapping source layer i to target layer i) is insufficient when the two models differ in depth. Following Li et al. (2025), we use Centred Kernel Alignment (CKA) (Kornblith et al., 2019) to measure the representational similarity between all pairs of layers across the two models, constructing a similarity matrix $\mathbf{S} \in \mathbb{R}^{L_{\text{src}} \times L_{\text{tgt}}}$ where $S_{i,j}$ quantifies how similar the representations of source layer i and target layer j are across a shared set of inputs. CKA is computed using the unbiased minibatch HSIC estimator of Nguyen et al. (2021), which makes the similarity values independent of batch size. A calibration corpus of 256 texts from WikiText-2[3] is used to collect the layer activations, running 32 minibatches through both models simultaneously.

Given $\mathbf{S}$, the optimal layer mapping is determined by a dynamic programming algorithm (Algorithm 2), which finds the strictly order-preserving assignment of source layers to target layers that maximises the total CKA similarity, subject to a maximum offset constraint Δ_{layer} that limits how far from the diagonal the mapping can deviate. The offset is set to 2 for same-family pairs (where layers are architecturally more similar) and to 4 for cross-family pairs (where greater divergence is expected). The CKA matrix is precomputed offline for each model pair and cached, so it doesn't need to be recomputed for each ported module.

A.3 Step 3: Attention Head Mapping

The third step handles mismatches in the number of attention heads between the source and target models. Within each mapped layer pair (i, j), the attention heads of the source layer are matched to those of the target layer using the Hungarian algorithm (Munkres, 1957), which finds the optimal one-to-one assignment minimising total assign-

[3] https://huggingface.co/datasets/Salesforce/wikitext

Algorithm 1: Pseudocode of LoRASuite portability pipeline, adapted from Li et al. (2025).

$\mathbf{B}_o, \mathbf{A}_o = \texttt{load_peft_weights}(\texttt{ORIGINAL_LORA_PATH})$
Step 1: Embedding transformation
$\mathbf{W}_h = \texttt{embedding_transform}(\text{ORIGINAL_TOKENIZER}, \text{NEW_TOKENIZER})$
Step 2: CKA-based layer mapping, where $S_{i,j}$ is the CKA similarity between layer i of the original model and layer j of the upgraded model
$L_{\text{dict}} = \texttt{cka_layer_mapping}(\mathbf{S})$
for *each* $(i,\ j)$ *in* L_{dict} **do**
  # Step 3: Attention head mapping, computing the Hungarian-based head mapping between the i-th projection layer of the original model $\mathbf{W}_{o,i}$ and the j-th layer of the new model $\mathbf{W}_{n,j}$
  $\Delta\mathbf{W}_{o,i} = \mathbf{B}_{o,i}\,\mathbf{A}_{o,i}$
  $H_{\text{dict}} = \texttt{hungarian_head_mapping}(\mathbf{W}_{o,i},\ \mathbf{W}_{n,j})$ **for** *each* $(h_o,\ h_n)$ *in* H_{dict} **do**
    $\Delta\mathbf{W}_{n,j}^{h_n} = \mathbf{W}_h^\top\,\Delta\mathbf{W}_{o,i}^{h_o}\,(\mathbf{W}_{o,i}^{h_o})^\top\,\mathbf{W}_h\,\mathbf{W}_{n,j}^{h_n}$
  # Step 4: LoRA weight transformation
  $\mathbf{B}_{n,j},\ \mathbf{A}_{n,j} = \texttt{SVD}(\Delta\mathbf{W}_{n,j})$

Algorithm 2: CKA-based Layer Mapping

Input:
(a) CKA similarity matrix $S \in \mathbb{R}^{l_o \times l_n}$
(b) original and upgrade layer depth l_o and l_n
Output: $path[i][j]$ with the highest total CKA

Initialise $dp[i][j] \leftarrow -\infty$ for all i, j
Initialise $path[i][j] \leftarrow 0$ for all i, j
for $j = 0$ ***to*** $|l_n - l_o|$ **do**
  $dp[0][j] \leftarrow S[0][j]$
for $i = 1$ ***to*** $l_o - 1$ **do**
  for $j = i$ ***to*** $i + |l_n - l_o|$ **do**
    $max_v \leftarrow -\infty$
    $max_i \leftarrow -1$
    for $k = i - 1$ ***to*** $j - 1$ **do**
      if $dp[i-1][k] + S[i][j] > max_v$ **then**
        $max_v \leftarrow dp[i-1][k] + S[i][j]$
        $max_i \leftarrow k$
    $dp[i][j] \leftarrow max_v$
    $path[i][j] \leftarrow max_i$

ment cost. Each attention head is characterised by two input-independent interaction matrices:

$$\begin{aligned}\mathbf{W}_{QK}^h &= \mathbf{W}_Q^h(\mathbf{W}_K^h)^\top,\\ \mathbf{W}_{VO}^h &= \mathbf{W}_V^h(\mathbf{W}_O^h)^\top\end{aligned} \tag{2}$$

where $\mathbf{W}_Q^h$, $\mathbf{W}_K^h$, $\mathbf{W}_V^h$, and $\mathbf{W}_O^h$ are the query, key, value, and output projection weights of head h respectively. $\mathbf{W}_{QK}^h$ captures the attention strength between token pairs, while $\mathbf{W}_{VO}^h$ models how attending to specific tokens influences the subsequent hidden state. When the hidden sizes differ across models, the interaction matrices are projected to a common space using $\mathbf{W}_h$ before computing cosine similarity. The resulting similarity matrix is passed to the Hungarian algorithm to obtain the optimal head-to-head assignment.

A.4 Step 4: LoRA Weight Transformation

Given the layer mapping from Step 2 and the head mapping from Step 3, the LoRA weights are transformed algebraically for each mapped layer pair $(i \to j)$ and each mapped head pair $(h_{\text{src}} \to h_{\text{tgt}})$. For attention projection layers, the weight update $\Delta\mathbf{W}$ for the source head is transformed to the target head's representation space as follows:

$$\begin{aligned}(\Delta\mathbf{W}^{h_{\text{tgt}}})_{\text{tgt}} = \mathbf{W}_h^\top \cdot (\Delta\mathbf{W}^{h_{\text{src}}})_{\text{src}} \cdot (\mathbf{W}^{h_{\text{src}}})_{\text{src}}^\top \\ \cdot\, \mathbf{W}_h \cdot (\mathbf{W}^{h_{\text{tgt}}})_{\text{tgt}}\end{aligned} \tag{3}$$

incorporating the original pretrained weights of both models to improve numerical stability. For MLP layers, the intermediate-dimension transformation $\mathbf{W}_i$ is applied analogously. After all layer and head transformations are completed, a single SVD step decomposes the transformed weight update $\Delta\mathbf{W}_{\text{tgt}}$ back into its low-rank factors $\mathbf{A}_{\text{tgt}}$ and $\mathbf{B}_{\text{tgt}}$, restoring the LoRA structure required for attachment to the target model. The ported module is then attached to the frozen target model and evaluated without any further training.

The complete portability pipeline is summarised in Algorithm 1. This training-free design is central to the plug-and-play philosophy of the framework: a module trained once on a source model can be ported to any compatible target model without access to the original training data or any gradient computation. In the few-step regime, the same algebraic transfer is performed first, followed by a short fine-tuning phase on the target model.

B Hyperparameters

LoRA hyperparameters. We set rank $r = 32$, scaling factor $\alpha = 64$, and dropout to 0.1. The down-projection matrix $\mathbf{A}$ is initialised using the Kaiming uniform distribution with $a = \sqrt{5}$, and the up-projection matrix $\mathbf{B}$ is initialised to all zeros. LoRA blocks are attached to all linear layers of the pretrained model, specifically the query, key, value, output, gate, up, and down projections.

Training. All modules are trained on a single NVIDIA A100 80GB GPU using the Paged AdamW optimiser (Loshchilov, 2017) with a cosine learning rate scheduler. The remaining hyperparameters are: learning rate 1×10^{-4}, batch size 16, gradient accumulation steps 8, maximum gradient norm 1.0, warmup ratio 0.05, weight decay 0.05, and maximum sequence length 1024 tokens. Training runs for up to 2 epochs, corresponding to 400 steps for sentiment control and 1,840 steps for topic control, reflecting the different sizes of the Yelp and AG News training sets. Rather than using the final checkpoint, we select the checkpoint with the lowest evaluation loss on the validation set.

Generation settings. Text is generated using greedy decoding (`do_sample=False`), with a repetition penalty of 1.3 and a no-repeat n-gram size of 4 to reduce degenerate repetition. Inputs are truncated to a maximum of 1024 tokens, and generation proceeds for up to 1024 new tokens per completion, stopping earlier if the model produces its end-of-sequence token. No top-p or top-k sampling is applied, as decoding is deterministic. Generated completions are post-processed by removing the `[ANS]` and `[\ANS]` tags before evaluation. We do not apply additional filtering for empty or degenerate outputs beyond this tag removal.

B.1 Evaluation Metrics

SLOR We used `gpt2-xl` and `bigscience/bloom-1b7` models from HuggingFace to compute sentence and unigram probabilities.

Control Effectiveness For sentiment control, we used `distilbert/distilbert-base-uncased-finetuned-sst-2-english`[4] and `michelecafagna26/t5-base-finetuned-sst2-sentiment`[5] from HuggingFace, and DeBERTa fine-tuned on Yelp (Gu et al., 2023). For topic control, we used `textattack/distilbert-base-uncased-ag-news`,[6] and `fabriceyhc/bert-base-uncased-ag_news`[7] from HuggingFace, and DeBERTa (Gu et al., 2023).

[4] https://huggingface.co/distilbert/distilbert-base-uncased-finetuned-sst-2-english

[5] https://huggingface.co/michelecafagna26/t5-base-finetuned-sst2-sentiment

[6] https://huggingface.co/textattack/distilbert-base-uncased-ag-news

[7] https://huggingface.co/fabriceyhc/bert-base-uncased-ag_news