GROWING AND TRANSFERRING PATHWAYS FOR LLM REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning (RL) can substantially improve the reasoning capabilities of large language models, but an effective training run requires considerable resources. Reusing the computation acquired through RL across models would extend the value of that training. We propose GrowRL, an RL method that organizes learned updates into compact computational pathways through structural growth. GrowRL freezes the pretrained backbone and adds an independent causal attention branch and a feed-forward network branch at each Transformer layer. Zero-initialized output projections preserve the pretrained policy at initialization, and RL updates only the added pathways. The learned pathways remain explicit after training, allowing transfer to structurally compatible target models and subsequent adaptation with the target backbone frozen. The architecture also supports sequential transfer of additional pathways. By keeping the learned attention and feed-forward computation separate from the pretrained weights, GrowRL provides an explicit structure for preserving and reusing the outcome of an RL run. Experiments across five backbones show competitive mathematical reasoning and out-of-distribution performance with a small number of trainable parameters. Learned pathways provide immediate gains across all three transfer directions, with performance after only 10 target-side updates already close to that after 100 updates in every case. These results support computational pathways as parameter-efficient and reusable units of reinforcement learning adaptation.

1 INTRODUCTION

Reinforcement learning with verifiable rewards (RLVR) has become an important approach for improving the reasoning performance of large language models (DeepSeek-AI et al., 2025; Liu et al., 2025). An effective RL run can produce substantial gains, but requires considerable training resources. Its value would extend beyond the model being trained if the computation acquired during that run could also benefit other models. Full-parameter reinforcement learning embeds updates throughout the trained backbone, while parameter-efficient methods preserve them in a smaller set of parameters (Houlsby et al., 2019; Hu et al., 2021; Liu et al., 2024). A separately stored update, however, is not necessarily useful in another model: the computation it performs must remain effective on representations produced by the receiving backbone. Reusable RL updates therefore face two linked requirements. They must improve the original model within a limited trainable budget and retain value after transfer to a compatible model.

We propose **GrowRL**, a reinforcement learning method that organizes learned updates into compact computational pathways. GrowRL keeps the pretrained backbone frozen and adds a trainable pathway to each Transformer layer (Vaswani et al., 2017). Each pathway contains an independent causal attention branch and a compact feed-forward network (FFN) branch, whose outputs are added to those of the corresponding original sublayers. The output projections of the new branches are zero-initialized, preserving the original model's behavior at the start of training. Reinforcement learning optimizes only the added pathways, leaving the learned components separate from the backbone parameters. After training, the pathways can be inserted into corresponding layers of a structurally compatible target model and further adapted there. Their attention and feed-forward structure provides a consistent interface for transferring learned computation between structurally compatible backbones.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

We evaluate GrowRL on multiple backbones and reasoning tasks, examining both learning on the original host and reuse in another model. On their original hosts, the pathways yield substantial reasoning improvements while training only a small number of added parameters, achieving performance competitive with full-parameter reinforcement learning across multiple settings. We then transfer trained pathways across three pairs of structurally compatible models and evaluate the receiving models before any target-side updates. In all three pairs, the transferred pathway improves MATH500 Pass@1 over the unmodified receiving model. Further reinforcement learning on the transferred pathway produces additional gains on Qwen2.5-1.5B and Qwen2.5-7B, while the pathway retains its initial advantage on Llama-3.2-3B. These results show that a pathway learned through one RL run can benefit its original host and remain useful after insertion into another compatible model.

GrowRL makes the computation acquired through reinforcement learning explicit in the model structure. The learned pathways can be retained independently of the original backbone and used as starting points for adaptation in a receiving model. The same structural design also permits further pathways to be added to an already adapted host. Together, these properties offer a way to preserve and reuse the outcome of an RL run across successive stages of model adaptation.

Our main contributions are as follows.

1. We propose **GrowRL**, which performs reinforcement learning through newly introduced attention and feed-forward pathways while keeping the pretrained backbone frozen. The learned updates remain identifiable components that can be inserted into structurally compatible models.
2. We show that compact GrowRL pathways can substantially improve reasoning performance on their original hosts and achieve performance competitive with full-parameter reinforcement learning across multiple settings.
3. We evaluate immediate cross-model transfer and subsequent target-side adaptation. Across three source–target pairs, transferred pathways improve MATH500 performance without target-side training and remain useful during further adaptation.

2 Related Work

2.1 Reinforcement Learning and Parameter-Efficient Adaptation

Reinforcement learning with verifiable rewards has become an important approach for improving the reasoning capabilities of large language models and has achieved substantial performance gains on mathematical and other reasoning tasks (DeepSeek-AI et al., 2025; Liu et al., 2025). Meanwhile, parameter-efficient adaptation has developed into a rich line of research, including adapters (Houlsby et al., 2019), Low-Rank Adaptation (LoRA) (Hu et al., 2021), Weight-Decomposed Low-Rank Adaptation (DoRA) (Liu et al., 2024), and other adaptation structures (He et al., 2021). These methods demonstrate that model updates can be concentrated within a relatively small parameter space. Related studies, including Tina, have also shown that a limited number of trainable parameters can effectively carry reinforcement learning updates and yield substantial improvements in reasoning performance (Wang et al., 2026). How trainable parameters are organized is therefore an important design dimension in reinforcement learning for model adaptation.

2.2 Modular Learning and Computational Structure Expansion

Modular learning focuses on preserving acquired capabilities in computational units with explicit structural boundaries. Adapters and their composition methods demonstrate the training and composition of independent modules (Houlsby et al., 2019; Pfeiffer et al., 2020). Progressive Networks acquire successive task capabilities, including in RL settings, by introducing new network structures (Rusu et al., 2016), while LLaMA-Pro increases computational capacity in existing language models by expanding Transformer blocks (Wu et al., 2024). Tiny-Attention Adapter introduces compact attention modules, providing another approach to adapting contextual computation (Zhao et al., 2022).

These studies establish that models can be adapted or expanded through newly introduced computational structures, with applications to task adaptation, module composition, and continual learning. GrowRL focuses on preserving and reusing the computation acquired through an RL run. It organizes reinforcement learning updates into attention and feed-forward computational pathways at each layer. These pathways remain structurally distinct from the frozen backbone after training, allowing the learned computation to be transferred to compatible models, further adapted, and incorporated into an already adapted host through sequential transfer.

3 GROWRL

GrowRL learns a computational pathway on a frozen source backbone and transfers the learned pathway to a structurally compatible target backbone. Let θ denote backbone parameters and ϕ pathway parameters. The pathway adds an attention branch and an FFN branch to each Transformer layer. Figure 1 illustrates source-model training, single transfer, and sequential transfer; A_θ, F_θ denote backbone sublayers, and A_i, F_i denote the branches of pathway i.

3.1 STRUCTURAL GROWTH AND SOURCE-MODEL TRAINING

Consider a pre-normalized decoder with L layers and hidden dimension d. At one layer, let $H \in \mathbb{R}^{T\times d}$ contain the residual states of T tokens. We suppress the layer index and denote the backbone attention and feed-forward sublayers by A_θ and F_θ, with normalizations N_A and N_F. GrowRL adds branches A_ϕ and F_ϕ that share the normalized inputs of the corresponding backbone sublayers:

$$\begin{aligned} X &= N_A(H), \quad U = H + A_\theta(X) + A_\phi(X), \\ Z &= N_F(U), \quad H' = U + F_\theta(Z) + F_\phi(Z). \end{aligned} \tag{1}$$

The added attention branch has its own query, key, value, and output projections. For all backbones considered here, it uses rotary positional embeddings (RoPE) (Su et al., 2021). With one head of dimension d_h, it computes

$$\begin{aligned} Q = \mathrm{RoPE}(XW_Q), \qquad K = \mathrm{RoPE}(XW_K), \qquad V = XW_V, \\ A_\phi(X) = \mathrm{softmax}\left(\frac{QK^\top}{\sqrt{d_h}} + M\right) VW_O. \end{aligned} \tag{2}$$

Here $W_Q, W_K, W_V \in \mathbb{R}^{d\times d_h}$, $W_O \in \mathbb{R}^{d_h\times d}$, and M masks future and padding positions. RoPE follows the backbone's positional convention, while the branch computes its own attention distribution. The FFN branch is a narrow SwiGLU transformation (Shazeer, 2020):

$$F_\phi(Z) = \big[\mathrm{SiLU}(ZW_g) \odot (ZW_u)\big] W_d. \tag{3}$$

Here $\mathrm{SiLU}(x) = x\,\sigma(x)$, where σ is the sigmoid function, and $W_g, W_u \in \mathbb{R}^{d\times r}$ and $W_d \in \mathbb{R}^{r\times d}$. We use $d_h = r = 128$ in the evaluated models. Both branches return to the d-dimensional residual stream, so subsequent backbone layers process the representations modified by the pathway.

Initialization and parameter budget. We initialize W_Q, W_K, W_V, W_g, W_u from $\mathcal{N}(0, 0.02^2)$ and set $W_O = W_d = 0$. The added branches therefore output zero initially, preserving the backbone's policy in exact arithmetic. Initially, only the output projections can receive nonzero loss gradients; gradients reach the internal projections after the output projections change (Appendix B.1). We train the added pathways while keeping the backbone projections, normalizations, embeddings, and language-model head fixed. The branch matrices contain $L(4dd_h + 3dr)$ parameters, giving approximately 38.5M trainable parameters for the 1.5B models.

Source-model reinforcement learning. We train the pathway using Dr. GRPO (Liu et al., 2025), sampling eight responses per prompt and assigning binary rewards from final-answer correctness. The backbone remains frozen, but its hidden states can change because later backbone layers process the pathway's outputs. GrowRL thus changes the trainable computational structure, while using the same reasoning objective and answer-verification reward as the comparison methods. The complete objective and implementation details appear in Appendices B.2 and B.3.

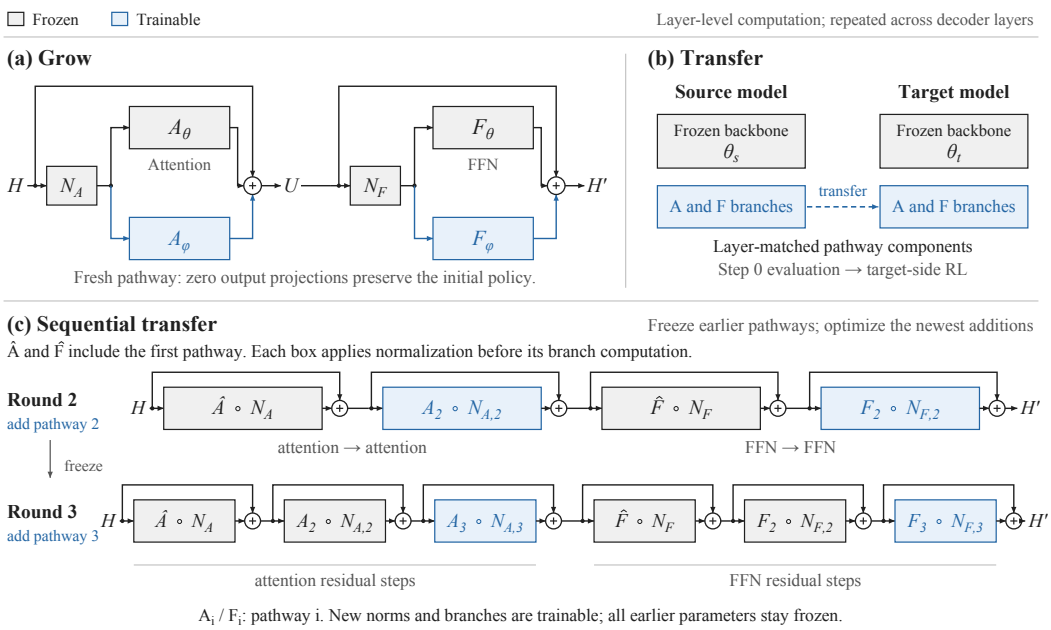

Figure 1: **GrowRL overview.** (a) Train a compact pathway on a frozen backbone. (b) Transfer the learned pathway to a compatible target model. (c) Insert further pathways through sequential transfer.

3.2 Transfer and Sequential Transfer

Transfer to a target backbone. After source-model training, we insert the learned pathway into the corresponding layers of a structurally compatible target backbone. Compatibility requires matching insertion sites and tensor dimensions for the transferred projections; no learned cross-model interface is used. The target retains its own embeddings, normalizations, attention and FFN parameters, and language-model head. Only pathway components are transferred. Transfer and target-side evaluation settings are given in Appendix A.3.

A trained pathway can change the target policy immediately upon insertion, without target-side optimization. We evaluate this zero-update model to measure immediate transfer. We then optionally optimize the inserted pathway through further RL while keeping the target backbone frozen. These two evaluations distinguish the benefit available from source-learned computation at insertion from improvements acquired through target-side adaptation. The pathway acts on representations produced by the receiving backbone, and target-side RL adapts its computation to this new host.

Sequential transfer to an adapted model. Sequential transfer inserts a source-trained pathway into a model that already contains a trained pathway. Let ϕ_1 be the first pathway, and let $\widehat{A}$ and $\widehat{F}$ denote the frozen attention and FFN computations of the adapted model, including ϕ_1. A second pathway ϕ_2 acts on the residual states produced by these adapted computations:

$$\begin{aligned} S &= H + \widehat{A}\big(N_A(H)\big), \\ U &= S + A_{\phi_2}\big(N_{A,2}(S)\big), \\ V &= U + \widehat{F}\big(N_F(U)\big), \\ H' &= V + F_{\phi_2}\big(N_{F,2}(V)\big). \end{aligned} \tag{4}$$

The resulting order is attention–attention–FFN–FFN. The additional root-mean-square normalization (RMSNorm; Zhang & Sennrich 2019) layers $N_{A,2}$ and $N_{F,2}$ are initialized by copying the corresponding normalizations and are trained with ϕ_2. Fresh pathways start with zero output projections, while transferred pathways load learned branch components. The new normalizations contribute $2Ld$ parameters. A third round adds another attention residual step after the second attention step and another FFN residual step after the second FFN step. All earlier parameters remain frozen.

The same sequential insertion structure also supports freshly initialized pathways. Sequential transfer uses source-trained components, while fresh pathways learn new components; both use the additional normalizations described above. Section 4.3 evaluates immediate and sequential transfer without target-side optimization, as well as post-transfer adaptation. Additional implementation details are given in Appendix B.3.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

We systematically evaluate GrowRL on language models of different sizes and types. Our experiments cover five backbones: FineMath-Llama-3B (Allal et al., 2025), Llama-3.2-3B (Meta, 2024), Qwen2.5-Math-1.5B and Qwen2.5-Math-7B (Yang et al., 2024), and Qwen2.5-7B (Qwen et al., 2024), spanning general-purpose pretrained models and mathematics-specialized models. We compare methods using shared training data, rewards, and evaluation protocols. Further experimental details are provided in Appendix A.1.

Reinforcement learning setup. We train with Dr. GRPO (Liu et al., 2025) with verifiable rewards. The training data consists of 5,585 Level 3–5 problems from the MATH training set (Hendrycks et al., 2021). We sample eight responses per problem and reward correct final answers. The standard source-model RL budget is 500 updates. In the post-transfer adaptation experiments, we further train the transferred components for 100 target-side updates while keeping the target backbone frozen. Immediate and sequential transfer are evaluated without target-side training. Default learning rates are 10^{-6} for Full RL and 10^{-5} for parameter-efficient methods. Complete settings are provided in Appendix A.1.

Comparison methods. We compare GrowRL with full-parameter RL (Full RL), LoRA (Hu et al., 2021), DoRA (Liu et al., 2024), and Parallel Adapter (PA) (He et al., 2021). Full RL updates all backbone parameters. LoRA and DoRA adapt the backbone projections, while PA adds a bottleneck branch alongside the FFN. GrowRL trains its added attention and FFN pathway. In immediate-transfer comparisons, Base denotes the unmodified target model and bold scores mark the best result across Base, LoRA, and GrowRL. The parameterizations are detailed in Appendix A.1.

Evaluation datasets. We evaluate models in terms of mathematical reasoning and out-of-distribution (OOD) generalization. The mathematical reasoning evaluation includes five datasets: MATH500 (Lightman et al., 2023), AMC2023 (AI-MO, 2024b), AIME2024 (AI-MO, 2024a), Minerva (Lewkowycz et al., 2022), and OlympiadBench (He et al., 2024), which assess reasoning across mathematical problems of varying difficulty and form. We further use six out-of-distribution datasets to evaluate generalization after mathematical reinforcement learning: GSM8K (Cobbe et al., 2021), HumanEval (Chen et al., 2021), MMLU-STEM and MMLU-Humanities (Hendrycks et al., 2020), ARC-Challenge (Clark et al., 2018), and HellaSwag (Zellers et al., 2019). These datasets cover mathematical and applied reasoning close to the training distribution, as well as more distant tasks involving code, knowledge, and general reasoning, allowing us to examine whether the capabilities acquired through mathematical reinforcement learning transfer beyond the training distribution.

Evaluation metrics. We consistently report sampled Pass@1 and Pass@8 (Chen et al., 2021) across all datasets. For each problem, we independently sample eight responses from the same generation distribution. Pass@1 reflects the average probability of success from a single sample, while Pass@8 measures the ability to obtain at least one correct answer among eight samples. Within each backbone comparison, all methods use the same prompts, sampling configurations, and answer-verification protocols. P1 and P8 denote Pass@1 and Pass@8. In the source-model RL tables, bold and underlined scores mark the best and second-best parameter-efficient results, respectively, with Full RL shown as a reference. Missing evaluations are left blank. The transfer-strength analysis uses four responses per problem on VAL300, as detailed in Appendix A.5.

Table 1: **Mathematical reasoning after source-model RL.** Sampled Pass@1/Pass@8 (%).

		MATH500		AMC2023		AIME2024		Minerva		Olympiad	
Backbone	**Method**	P1	P8	P1	P8	P1	P8	P1	P8	P1	P8
FineMath-Llama-3B	Full RL	34.65	65.00	14.06	40.00	0.42	3.33	14.02	33.09	9.51	28.04
	LoRA	17.65	51.00	5.94	25.00	0.00	0.00	6.57	26.84	3.19	14.84
	DoRA	7.92	35.20	2.50	20.00	0.00	0.00	3.63	18.75	1.85	11.42
	PA	30.58	60.60	11.25	**40.00**	0.42	**3.33**	11.86	**35.29**	7.18	23.89
	GrowRL	**33.30**	**63.00**	**12.50**	**40.00**	**1.67**	**3.33**	**12.59**	32.72	**8.92**	**26.26**
Llama-3.2-3B	Full RL	10.35	19.60	5.63	17.50	0.00	0.00	9.05	19.85	3.39	6.23
	LoRA	5.53	25.20	3.75	22.50	0.00	0.00	4.09	17.65	1.97	10.83
	DoRA	3.60	19.60	5.00	**35.00**	0.00	0.00	2.07	10.29	1.13	7.12
	PA	9.25	**34.20**	**6.88**	25.00	**0.83**	**6.67**	**8.23**	**23.16**	**3.65**	**13.50**
	GrowRL	**9.73**	24.20	5.94	25.00	0.00	0.00	8.04	21.32	3.39	10.39
Qwen2.5-Math-1.5B	Full RL	65.05	85.20	42.81	77.50	14.17	30.00	21.60	36.76	26.93	44.66
	LoRA	60.75	**85.60**	43.44	72.50	**9.58**	26.67	18.52	35.66	25.24	**45.99**
	DoRA	57.33	82.60	41.88	72.50	8.33	**30.00**	17.51	35.29	23.33	43.47
	PA	46.80	82.20	32.81	**75.00**	5.00	23.33	13.10	**37.87**	18.99	42.43
	GrowRL	**63.20**	85.00	**45.94**	**75.00**	9.17	26.67	**19.44**	36.76	**26.67**	44.51
Qwen2.5-Math-7B	Full RL	67.00	88.40	58.44	75.00	19.58	50.00	23.30	47.06	29.51	48.52
	LoRA	58.00	85.60	47.50	82.50	15.83	46.67	18.43	44.49	24.93	46.88
	DoRA	55.70	87.20	47.19	**85.00**	13.75	40.00	16.18	41.18	23.22	46.29
	PA	**68.62**	**88.80**	**59.69**	82.50	**21.25**	**53.33**	**24.36**	**45.22**	**30.86**	47.92
	GrowRL	63.75	87.20	54.06	82.50	18.75	43.33	19.49	43.75	27.89	**48.66**
Qwen2.5-7B	Full RL	74.98	88.80	57.50	85.00	10.42	26.67	29.37	43.38	35.57	52.67
	LoRA	65.68	86.40	47.50	72.50	12.50	**30.00**	25.37	**46.69**	28.26	46.88
	DoRA	57.80	84.40	31.87	72.50	6.67	20.00	18.84	42.65	23.59	44.96
	PA	70.10	85.80	51.56	**80.00**	12.92	26.67	26.61	43.75	32.64	**49.26**
	GrowRL	**72.95**	**87.60**	**55.00**	72.50	**15.83**	**30.00**	**28.03**	44.85	**32.94**	48.52

4.2 Source-Model Reinforcement Learning

GrowRL remains consistently competitive across a broad range of models and tasks. Tables 1 and 2 summarize source-model RL results for five backbones across 11 datasets. GrowRL achieves competitive performance in both mathematical reasoning and out-of-distribution evaluation, maintaining strong results under both Pass@1 and Pass@8. The best parameter-efficient method varies across backbones and metrics. On Qwen2.5-Math-7B, GrowRL exceeds LoRA in Pass@1 on all five mathematical benchmarks, while PA leads the parameter-efficient methods on these benchmarks under this metric. Overall, these results suggest that structural growth is effective beyond a small number of specific settings and can serve as a broadly applicable approach to reinforcement learning adaptation.

Structural growth approaches full-parameter reinforcement learning performance with a small number of additional parameters. Full RL updates all backbone parameters, whereas GrowRL optimizes only the added computational pathways. Despite a substantially smaller trainable parameter space, GrowRL achieves performance close to Full RL in many settings and matches or surpasses Full RL on some tasks and metrics. This suggests that the effective plasticity required for reinforcement learning need not be distributed throughout the entire pretrained model: compact, added computational pathways can also support substantial policy improvements.

Parameter-efficient reinforcement learning depends not only on how many parameters are trained, but also on how they are organized. LoRA, DoRA, PA, and GrowRL all restrict reinforcement learning to a small trainable parameter space, yet their performance differs substantially across backbones and tasks. GrowRL organizes its trainable parameters into complete pathways with independent attention and feed-forward computation, with competitive performance across multiple settings. These results suggest that the structural organization of trainable capacity is itself a key design dimension in parameter-efficient reinforcement learning, beyond the number of trainable parameters alone.

GrowRL maintains strong reasoning capabilities beyond the training distribution. Although all models undergo reinforcement learning primarily on mathematical reasoning data, GrowRL maintains strong performance on out-of-distribution tasks including GSM8K, HumanEval, MMLU-

Table 2: **OOD performance after source-model RL.** Sampled Pass@1/Pass@8 (%).

		GSM8K		HumanEval		MMLU-STEM		ARC-C		MMLU-H		HellaSwag	
Backbone	**Method**	P1	P8	P1	P8	P1	P8	P1	P8	P1	P8	P1	P8
FineMath-Llama-3B	Full RL	59.44	87.34	0.00	0.00	12.80	62.69	15.66	72.10	12.81	63.38	12.17	56.87
	LoRA	31.25	75.44	0.00	0.00	11.56	61.03	14.05	68.00	11.96	60.60	12.15	**57.07**
	DoRA	11.43	49.96	**0.08**	**0.61**	11.03	58.48	12.81	63.65	10.94	56.47	11.60	54.81
	PA	54.43	86.88	0.00	0.00	12.27	60.80	15.18	68.77	12.89	63.34	**12.26**	56.70
	GrowRL	**59.51**	**87.34**	0.00	0.00	**15.12**	**66.20**	**19.64**	**75.26**	**15.06**	**67.74**	**12.26**	57.01
Llama-3.2-3B	Full RL	15.57	38.21	19.97	44.51	7.14	40.06	8.75	46.25	11.71	55.01	7.79	42.79
	LoRA	8.48	38.59	13.64	37.20	8.81	**49.90**	10.65	54.86	11.21	57.07	8.91	**48.13**
	DoRA	4.30	26.69	11.59	34.76	8.19	47.68	9.82	52.82	10.71	55.83	8.62	46.73
	PA	13.62	47.84	18.60	**42.68**	2.92	18.95	4.58	27.90	5.41	32.75	5.78	32.83
	GrowRL	**16.62**	**48.22**	**20.20**	40.85	**10.35**	49.87	**15.31**	**62.54**	**13.61**	**59.26**	**9.46**	44.16
Qwen2.5-Math-1.5B	Full RL	77.23	94.47	24.31	68.29	44.57	83.80	53.30	93.43	31.89	84.65	11.84	45.54
	LoRA	72.65	94.31	20.58	65.85	**43.85**	85.02	49.25	**91.89**	30.29	84.21	**14.02**	53.27
	DoRA	68.85	94.47	6.10	32.32	41.96	84.86	45.49	91.64	28.55	**84.72**	13.71	**54.02**
	PA	55.93	93.63	5.79	34.76	32.67	**86.78**	32.35	89.33	21.49	80.70	11.84	51.76
	GrowRL	**74.72**	**95.22**	**31.10**	**70.73**	43.18	81.58	**51.75**	**91.89**	**31.25**	83.66	11.90	46.84
Qwen2.5-Math-7B	Full RL	81.43	97.80	35.82	76.22	33.55	88.97	38.32	94.28	25.57	83.36	14.44	58.43
	LoRA	70.50	97.35	29.95	**75.61**	37.88	**90.92**	43.76	94.88	26.53	84.87	16.53	66.17
	DoRA	67.39	96.82	28.05	72.56	33.03	89.33	41.80	94.88	25.23	84.27	15.49	64.50
	PA	**80.35**	**97.50**	29.12	73.17	**55.48**	89.76	**64.74**	**95.90**	**38.96**	**86.95**	**18.80**	**68.28**
	GrowRL	76.30	**97.50**	**30.95**	73.78	34.05	89.43	40.38	93.94	25.62	84.25	16.47	65.24
Qwen2.5-7B	Full RL	91.60	96.89	69.74	91.46	68.07	86.61	87.62	96.42	58.59	84.95	40.56	85.71
	LoRA	85.08	97.12	**54.50**	**90.85**	66.09	92.78	79.76	96.84	51.27	86.82	28.19	**81.15**
	DoRA	78.71	**97.50**	26.37	79.88	56.57	**93.64**	69.39	**97.35**	43.71	**87.23**	26.59	81.11
	PA	**90.51**	96.89	15.17	60.98	**68.06**	90.39	**81.76**	95.90	**54.93**	82.95	**29.26**	79.13
	GrowRL	90.38	97.27	53.43	86.59	63.56	89.46	80.32	96.84	51.89	86.91	21.73	70.96

STEM, ARC-Challenge, MMLU-Humanities, and HellaSwag, matching or surpassing Full RL in multiple settings. This suggests that the computational pathways learned by GrowRL are not limited to the mathematical training distribution, but can support reinforcement learning updates that are also valuable for code, knowledge, scientific reasoning, and general reasoning.

Structural growth remains effective across model types and scales. The five backbones cover both general-purpose language models and mathematics-specialized models, spanning scales of 1.5B, 3B, and 7B parameters. Despite substantial differences in their pretraining objectives, initial capabilities, and domain biases, the same GrowRL structure yields effective reinforcement learning results on all models. This indicates that the effectiveness of structural growth does not depend on a particular backbone and extends across model types and scales.

4.3 Transfer and Rapid Adaptation

We evaluate immediate transfer to structurally compatible targets, post-transfer adaptation, and sequential transfer of source-trained pathways. The target backbone remains frozen throughout; immediate and sequential transfer use no target-side training.

Transferring computational pathways yields substantial immediate gains that extend beyond mathematical tasks. Table 3 shows that GrowRL improves MATH500 performance in all three transfer directions without target-side training, outperforming direct LoRA transfer. Gains also extend to GSM8K, HumanEval, MMLU-STEM, ARC-Challenge, and MMLU-Humanities. Thus, pathways learned through mathematical RL retain value across receiving models and beyond the source training distribution.

Transferred computational pathways adapt rapidly to target models with only a few reinforcement learning updates. In Figure 2, GrowRL's MATH500 Pass@1 after 10 target-side updates is already close to its performance at update 100 on Qwen2.5-1.5B and Llama-3.2-3B, and it remains above LoRA at the measured checkpoints. Qwen2.5-1.5B improves clearly within the first 10 updates, while Llama-3.2-3B retains its immediate transfer gains. Learned pathways therefore provide an effective starting point for adaptation with a frozen target backbone.

Table 3: **Immediate transfer performance.** Sampled Pass@1 (%) without target-side training.

Method	MATH	AMC	AIME	Minerva	Olymp.	GSM8K	HEval	STEM	ARC-C	MMLU-H
Qwen2.5-Math-1.5B → Qwen2.5-1.5B										
Base	4.63	2.50	0.00	1.10	1.13	6.12	3.28	6.58	7.60	6.25
LoRA	6.90	3.44	0.00	2.25	1.69	9.04	6.33	7.30	8.44	6.71
GrowRL	**22.63**	**10.94**	**1.67**	**4.78**	**7.07**	**26.08**	**12.27**	**9.63**	**10.61**	**7.31**
FineMath-Llama-3B → Llama-3.2-3B										
Base	3.88	2.19	**0.00**	2.07	1.11	4.04	10.82	8.34	10.14	10.94
LoRA	4.45	**3.44**	**0.00**	3.63	1.39	7.15	14.63	9.35	10.63	11.85
GrowRL	**9.63**	2.81	**0.00**	**5.84**	**2.26**	**15.57**	**22.26**	**9.72**	**16.21**	**15.16**
Qwen2.5-Math-7B → Qwen2.5-7B										
Base	32.08	20.31	**3.75**	8.87	10.89	48.54	24.92	36.44	46.76	28.66
LoRA	34.78	20.31	3.33	10.02	10.92	51.92	22.41	41.67	52.22	32.49
GrowRL	**40.88**	**20.94**	**3.75**	**13.56**	**14.22**	**54.52**	**38.64**	**48.67**	**61.77**	**39.30**

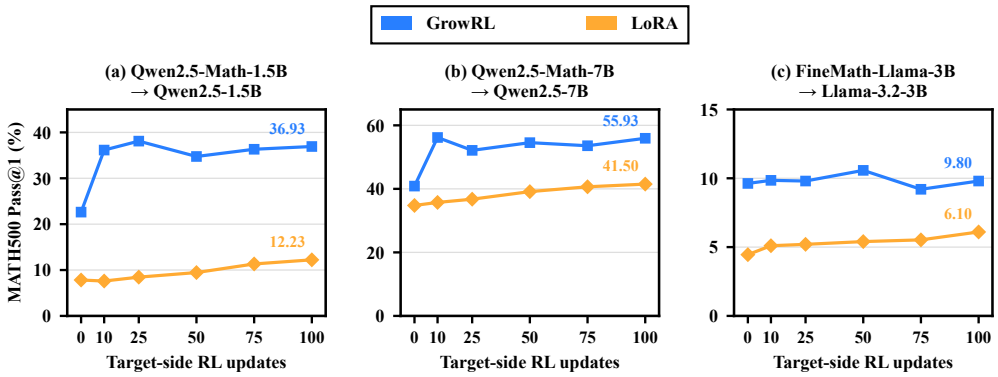

Figure 2: **Rapid adaptation after transfer.** MATH500 Pass@1 with a frozen target backbone.

Sequential transfer can yield further gains without target-side training. Starting from each Base target, we perform up to four sequential transfers using pathway components from the same source checkpoint, without target-side optimization. Table 4 reports the complete sweep, with 0 denoting the unmodified target. Qwen2.5-1.5B peaks at two transfers, improving Pass@1 from 23.03% after one transfer to 42.15%. Qwen2.5-7B improves from 39.53% to 53.28% over one to four transfers. Llama-3.2-3B instead peaks at one transfer and declines thereafter. Sequential transfer can thus extract additional value from source-learned computation, with the useful number of transfers depending on the receiving model.

Table 4: **Sequential transfer.** MATH500 Pass@1/Pass@8 (%) without target-side training.

	Qwen2.5-Math-1.5B ↓ Qwen2.5-1.5B		Qwen2.5-Math-7B ↓ Qwen2.5-7B		FineMath-Llama-3B ↓ Llama-3.2-3B	
Transfers	P1	P8	P1	P8	P1	P8
0 (Base)	4.63	27.80	31.50	76.60	4.00	21.20
1	23.03	62.80	39.53	80.00	**9.50**	**31.20**
2	**42.15**	**75.20**	46.08	81.60	4.93	15.00
3	41.23	71.20	48.93	80.40	4.75	13.60
4	33.40	62.80	**53.28**	**82.20**	4.55	10.80

Table 5: **Effect of trainable pathway components.** Sampled Pass@1 (%) after source-model RL.

Backbone	Trainable part	Params (M)	MATH500	AMC2023	AIME2024	Minerva	Olympiad
Qwen2.5-Math-1.5B	Full pathway	38.54	**63.20**	**45.94**	**9.17**	**19.44**	**26.67**
	W_O, W_d only	11.01	50.75	33.13	7.92	14.57	19.42
Qwen2.5-Math-7B	Full pathway	89.92	**63.75**	**54.06**	18.75	19.49	**27.89**
	W_O, W_d only	25.69	63.15	**54.06**	**19.17**	**21.23**	27.60
FineMath-Llama-3B	Full pathway	77.07	**33.30**	**12.50**	**1.67**	**12.59**	**8.92**
	W_O, W_d only	22.02	11.53	3.44	0.00	4.96	2.47

4.4 Fine-Grained Analysis

Training internal transformations can improve pathway adaptation. Table 5 compares *Full pathway*, which trains all added projections, with W_O, W_d *only*, which freezes the internal projections. The latter trains only the attention output and FFN down projections that return branch features to the residual stream. Full-pathway training improves all five mathematical benchmarks on Qwen2.5-Math-1.5B and FineMath-Llama-3B. On Qwen2.5-Math-7B, it improves MATH500 and OlympiadBench, underperforms the two-projection variant on AIME2024 and Minerva, and ties on AMC2023. Learning internal transformations can therefore improve adaptation, with benefits that depend on the backbone and task.

Increasing the contribution of learned pathways improves overall performance after transfer. We vary a fixed global coefficient $\alpha \in \{0, 0.05, 0.1, 0.2, 0.5, 1\}$ at evaluation time, without target-side training. For GrowRL, the same α scales both transferred branch outputs at every layer:

$$A_\phi^{(\alpha)}(X) = \alpha A_\phi(X), \qquad F_\phi^{(\alpha)}(Z) = \alpha F_\phi(Z). \tag{5}$$

These scaled outputs replace $A_\phi(X)$ and $F_\phi(Z)$ in Equation (1). For LoRA, each adapted target projection is $W_t^{(\alpha)} = W_t + \alpha \Delta W$, where ΔW is the transferred update including its original adapter scaling. All learned parameters remain fixed: $\alpha = 0$ recovers the unmodified target model, and $\alpha = 1$ applies the full transferred contribution. Figure 3 shows that GrowRL's sampled Pass@1 on VAL300 generally increases with α, with a substantially larger gain than LoRA. Local changes are not strictly monotonic, but full strength performs best among the evaluated settings, supporting the contribution of the learned pathways to transfer gains.

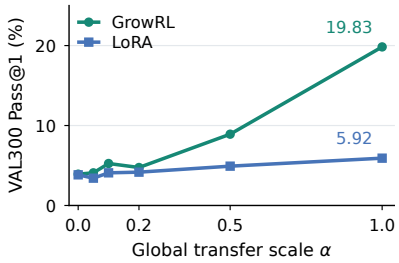

Figure 3: **Effect of transfer strength.** VAL300 Pass@1 for Qwen2.5-Math-1.5B → Qwen2.5-1.5B.

5 Conclusion

GrowRL learns compact attention and feed-forward pathways on frozen backbones. Across five backbones, it achieves competitive mathematical and out-of-distribution performance, while transferred pathways provide immediate gains and rapid adaptation. These findings show that the computation acquired through an expensive RL run need not remain inseparably tied to a single model. By making this computation explicit, modular, and transferable, GrowRL offers a step toward reusable units of RL adaptation.

AI USE STATEMENT

In this work, generative AI tools, including OpenAI Codex, were used to polish the language, assist with literature research, and support experimental reproduction, implementation, and manuscript formatting. They were not involved in formulating the core research ideas or designing the central method. All AI-assisted text, code, experimental procedures, and literature-related suggestions were reviewed and verified by the authors. The authors take full responsibility for the final content of this work, including all claims and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Reproducibility is supported by the pathway architecture, initialization, transfer, and sequential-transfer specification in Section 3; the complete RL objective and implementation details in Appendix B; and the training budgets, adaptation methods, evaluation protocols, target checkpoints, and ablation settings in Appendix A.

REFERENCES

AI-MO. AIMO Validation AIME. Hugging Face dataset, 2024a. URL https://huggingface.co/datasets/AI-MO/aimo-validation-aime.

AI-MO. AIMO Validation AMC. Hugging Face dataset, 2024b. URL https://huggingface.co/datasets/AI-MO/aimo-validation-amc.

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. GQA: Training Generalized Multi-Query Transformer Models from Multi-Head Checkpoints. *arXiv preprint arXiv:2305.13245*, 2023. URL https://arxiv.org/abs/2305.13245.

Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlíček, Agustín Piqueres Lajarín, Vaibhav Srivastav, Joshua Lochner, Caleb Fahlgren, Xuan-Son Nguyen, Clémentine Fourrier, Ben Burtenshaw, Hugo Larcher, Haojun Zhao, Cyril Zakka, Mathieu Morlon, Colin Raffel, Leandro von Werra, and Thomas Wolf. SmolLM2: When Smol Goes Big – Data-Centric Training of a Small Language Model. *arXiv preprint arXiv:2502.02737*, 2025. URL https://arxiv.org/abs/2502.02737.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, et al. Evaluating Large Language Models Trained on Code. *arXiv preprint arXiv:2107.03374*, 2021. URL https://arxiv.org/abs/2107.03374.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have Solved Question Answering? Try ARC, the AI2 Reasoning Challenge. *arXiv preprint arXiv:1803.05457*, 2018. URL https://arxiv.org/abs/1803.05457.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training Verifiers to Solve Math Word Problems. *arXiv preprint arXiv:2110.14168*, 2021. URL https://arxiv.org/abs/2110.14168.

DeepSeek-AI et al. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. *arXiv preprint arXiv:2501.12948*, 2025. URL https://arxiv.org/abs/2501.12948.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. OlympiadBench: A Challenging Benchmark for Promoting AGI with Olympiad-Level Bilingual Multimodal Scientific Problems. *arXiv preprint arXiv:2402.14008*, 2024. URL https://arxiv.org/abs/2402.14008.

Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. Towards a Unified View of Parameter-Efficient Transfer Learning. *arXiv preprint arXiv:2110.04366*, 2021. URL https://arxiv.org/abs/2110.04366.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring Massive Multitask Language Understanding. *arXiv preprint arXiv:2009.03300*, 2020. URL https://arxiv.org/abs/2009.03300.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring Mathematical Problem Solving With the MATH Dataset. *arXiv preprint arXiv:2103.03874*, 2021. URL https://arxiv.org/abs/2103.03874.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-Efficient Transfer Learning for NLP. *arXiv preprint arXiv:1902.00751*, 2019. URL https://arxiv.org/abs/1902.00751.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models. *arXiv preprint arXiv:2106.09685*, 2021. URL https://arxiv.org/abs/2106.09685.

Hugging Face. PEFT troubleshooting: Selecting the dtype of the adapter, 2026a. URL https://huggingface.co/docs/peft/developer_guides/troubleshooting. Documentation, accessed September 26, 2026.

Hugging Face. TRL: GRPO trainer, 2026b. URL https://huggingface.co/docs/trl/grpo_trainer. Documentation, accessed September 26, 2026.

Hugging Face. TRL: PEFT integration, 2026c. URL https://huggingface.co/docs/trl/peft_integration. Documentation, accessed September 26, 2026.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. Solving Quantitative Reasoning Problems with Language Models. *arXiv preprint arXiv:2206.14858*, 2022. URL https://arxiv.org/abs/2206.14858.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's Verify Step by Step. *arXiv preprint arXiv:2305.20050*, 2023. URL https://arxiv.org/abs/2305.20050.

Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. DoRA: Weight-Decomposed Low-Rank Adaptation. *arXiv preprint arXiv:2402.09353*, 2024. URL https://arxiv.org/abs/2402.09353.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding R1-Zero-Like Training: A Critical Perspective. *arXiv preprint arXiv:2503.20783*, 2025. URL https://arxiv.org/abs/2503.20783.

Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. *arXiv preprint arXiv:1711.05101*, 2017. URL https://arxiv.org/abs/1711.05101.

Meta. Llama 3.2 Model Card. Official model documentation, 2024. URL https://github.com/meta-llama/llama-models/blob/main/models/llama3_2/MODEL_CARD.md.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. AdapterFusion: Non-Destructive Task Composition for Transfer Learning. *arXiv preprint arXiv:2005.00247*, 2020. URL https://arxiv.org/abs/2005.00247.

Qwen et al. Qwen2.5 Technical Report. *arXiv preprint arXiv:2412.15115*, 2024. URL https://arxiv.org/abs/2412.15115.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. ZeRO: Memory Optimizations Toward Training Trillion Parameter Models. *arXiv preprint arXiv:1910.02054*, 2019. URL https://arxiv.org/abs/1910.02054.

Andrei A. Rusu, Neil C. Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive Neural Networks. *arXiv preprint arXiv:1606.04671*, 2016. URL https://arxiv.org/abs/1606.04671.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms. *arXiv preprint arXiv:1707.06347*, 2017. URL https://arxiv.org/abs/1707.06347.

Noam Shazeer. GLU Variants Improve Transformer. *arXiv preprint arXiv:2002.05202*, 2020. URL https://arxiv.org/abs/2002.05202.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. RoFormer: Enhanced Transformer with Rotary Position Embedding. *arXiv preprint arXiv:2104.09864*, 2021. URL https://arxiv.org/abs/2104.09864.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. *arXiv preprint arXiv:1706.03762*, 2017. URL https://arxiv.org/abs/1706.03762.

Shangshang Wang, Julian Asilis, Ömer Faruk Akgül, Enes Bilgin, Ollie Liu, and Willie Neiswanger. Tina: Tiny Reasoning Models via LoRA. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/91be140e1524bb21531508c2a6a85c5e-Abstract-Conference.html.

Chengyue Wu, Yukang Gan, Yixiao Ge, Zeyu Lu, Jiahao Wang, Ye Feng, Ying Shan, and Ping Luo. LLaMA Pro: Progressive LLaMA with Block Expansion. *arXiv preprint arXiv:2401.02415*, 2024. URL https://arxiv.org/abs/2401.02415.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. Qwen2.5-Math Technical Report: Toward Mathematical Expert Model via Self-Improvement. *arXiv preprint arXiv:2409.12122*, 2024. URL https://arxiv.org/abs/2409.12122.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a Machine Really Finish Your Sentence? *arXiv preprint arXiv:1905.07830*, 2019. URL https://arxiv.org/abs/1905.07830.

Biao Zhang and Rico Sennrich. Root Mean Square Layer Normalization. *arXiv preprint arXiv:1910.07467*, 2019. URL https://arxiv.org/abs/1910.07467.

Hongyu Zhao, Hao Tan, and Hongyuan Mei. Tiny-Attention Adapter: Contexts Are More Important Than the Number of Parameters. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 6626–6638. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.emnlp-main.444. URL https://aclanthology.org/2022.emnlp-main.444/.

A Experimental Details and Additional Results

A.1 Source-model training

The source backbones are FineMath-Llama-3B, Llama-3.2-3B, Qwen2.5-Math-1.5B, Qwen2.5-Math-7B, and Qwen2.5-7B. Qwen2.5-1.5B (Qwen et al., 2024) is also a transfer target. The standard source-model RL budget is 500 updates. Post-transfer adaptation trains only the transferred components for 100 target-side updates with the target backbone frozen. Immediate and sequential transfer use no target-side training.

Training uses 5,585 Level 3–5 problems from the MATH training set, with 84 prompts and eight responses per prompt per update (672 responses), temperature 1, and top-$p = 1$. Binary rewards indicate final-answer correctness. We use Dr. GRPO with group-centered rewards, no reward-standard-deviation normalization, and no KL penalty. The maximum completion length is 3,000 tokens and the model context is 4,096 tokens.

We use AdamW (Loshchilov & Hutter, 2017) with $(\beta_1, \beta_2) = (0.9, 0.95)$, $\epsilon = 10^{-8}$, gradient clipping at 1, and no weight decay or warmup. The default learning rates are 10^{-6} for Full RL and 10^{-5} for parameter-efficient methods. This choice follows common GRPO practice: the TRL guidelines recommend 10^{-6} for full fine-tuning and 10^{-5} for LoRA (Hugging Face, 2026c). Forward and backward computation predominantly use bfloat16 (BF16), while master parameters and Adam optimizer states use single precision (FP32). This mixed-precision setup reduces memory and compute costs while retaining higher-precision optimizer updates; BF16 is also the standard precision in the TRL GRPO configuration (Hugging Face, 2026b). We distribute optimizer states and gradients with Zero Redundancy Optimizer stage 2 (ZeRO-2) (Rajbhandari et al., 2019).

Adaptation methods. Full RL updates all backbone parameters. LoRA applies rank-32 updates to the query, key, value, attention output, and three FFN projections, with adapter scaling $\alpha_{\mathrm{LoRA}}/r = 1$ and no adapter dropout. DoRA uses the same rank and target projections and additionally trains weight magnitudes. PA uses a width-192 bottleneck with a rectified linear unit (ReLU) activation alongside the FFN; its output projection is zero-initialized. PA is a comparison method, distinct from GrowRL's sequential-transfer construction.

GrowRL adds one attention head of width 128 and one SwiGLU branch of width 128 per layer. Both output projections are zero-initialized. The backbone remains frozen. The pathway has approximately 38.5M, 77.1M, and 89.9M trainable parameters for the 1.5B, 3B, and 7B backbones, respectively.

Precision and the scope of the DoRA baseline. DoRA learns a weight magnitude as well as a low-rank direction (Liu et al., 2024). We evaluate it under the same BF16 computation and FP32 master-parameter and optimizer-state policy as the other methods. Our checkpoint audit found that DoRA's BF16 magnitude parameters changed little or not at all despite updates to their FP32 master copies, indicating that small magnitude updates were lost during conversion to BF16. PEFT normally promotes adapter parameters to FP32 where appropriate (Hugging Face, 2026a), whereas our shared configuration uses BF16 for forward computation. The reported results characterize DoRA under this common training setup, rather than its best performance with a separately optimized precision configuration.

Execution. Experiments use NVIDIA A800 GPUs with 80 GB of memory. Training is implemented with PyTorch, Transformers, TRL, PEFT, and DeepSpeed; vLLM generates rollouts and evaluation responses. Gradient accumulation preserves the effective prompt batch when the number of training GPUs changes. Evaluation uses tensor parallelism of one.

A.2 Evaluation protocols

The mathematical evaluation uses all 500 MATH500 problems (Lightman et al., 2023) drawn from the MATH benchmark (Hendrycks et al., 2021), 40 AMC2023 problems (AI-MO, 2024b), 30 AIME2024 problems (AI-MO, 2024a), 272 Minerva problems (Lewkowycz et al., 2022), and 674 problems from the English text-only mathematics subset of OlympiadBench (He et al.,

2024). The OOD evaluation uses 1,319 GSM8K questions (Cobbe et al., 2021), 164 HumanEval tasks (Chen et al., 2021), 3,018 MMLU-STEM and 4,705 MMLU-Humanities questions (Hendrycks et al., 2020), 1,172 ARC-Challenge questions (Clark et al., 2018), and 10,042 HellaSwag examples (Zellers et al., 2019).

For the source-model RL and immediate-transfer comparisons, each problem receives eight sampled responses at temperature 1 and top-$p = 1$. Pass@1 is the mean correctness of these responses, averaged over problems. Pass@8 is the fraction of problems with at least one correct response. It does not use majority voting. Both metrics are reported as percentages.

Qwen2.5-Math-1.5B, Qwen2.5-Math-7B, Qwen2.5-1.5B, and Qwen2.5-7B use a boxed-answer prompt for mathematical evaluation. FineMath-Llama-3B and Llama-3.2-3B use the R1-style prompt with the `</answer>` stop sequence. Methods within a backbone comparison share the prompt and decoding protocol. Mathematical answers are checked with answer verification, HumanEval programs with execution-based tests, and multiple-choice tasks with answer-letter extraction. Completion limits are 3,000 tokens for the five mathematical benchmarks, 2,048 for GSM8K, 1,024 for HumanEval, 512 for MMLU and ARC-Challenge, and 64 for HellaSwag.

Table headings use MATH for MATH500, AMC for AMC2023, AIME for AIME2024, Olymp. for OlympiadBench, HEval for HumanEval, STEM for MMLU-STEM, ARC-C for ARC-Challenge, and MMLU-H for MMLU-Humanities. These abbreviations refer to the same datasets throughout the paper.

A.3 Transfer and post-transfer adaptation

We study Qwen2.5-Math-1.5B $\rightarrow$ Qwen2.5-1.5B, Qwen2.5-Math-7B $\rightarrow$ Qwen2.5-7B, and FineMath-Llama-3B $\rightarrow$ Llama-3.2-3B. The GrowRL source pathways are taken from update 500. The LoRA adapter is held fixed across the corresponding immediate-transfer Pass@1 and Pass@8 comparisons. Immediate transfer uses zero target updates. Post-transfer adaptation runs for 100 target updates with a fresh optimizer and a frozen target backbone; only the transferred parameters are trained.

LoRA transfers the complete rank-32 adapter at its original scale. Evaluation uses the same MATH500 protocol as the source-model RL comparison. Table 6 lists the measured target checkpoints; line segments in Figure 2 connect these observations. The Qwen2.5-1.5B LoRA trajectory uses an independent eight-response evaluation of the same initial adapter weights, so its sampled update-0 score can differ from the immediate-transfer table.

Sequential transfer. Table 4 uses the same three source–target directions, starting from each unmodified Base target. We evaluate zero through four cumulative transfers, using pathway components from the same fixed source checkpoint for each source–target pair. After the first transfer, each additional attention branch follows the preceding attention computations and each additional FFN branch follows the preceding FFN computations, as in Equation (4). The additional normalizations copy the corresponding target RMSNorm weights. All parameters remain fixed, with no target-side optimization between or after transfers. Each configuration is evaluated on all 500 MATH500 problems with eight responses, temperature 1, top-$p = 1$, a 3,000-token completion limit, a 4,096-token context, and the target-specific prompt described above. This sweep uses independent sampling, so its zero- and one-transfer scores can differ from Table 3. Bold entries mark the highest score across transfer counts for each target and metric.

A.4 Additional immediate-transfer results

Table 7 complements the Pass@1 comparison in Table 3. Base denotes the unmodified target model. Each Pass@8 entry is computed from the same eight responses per problem as its corresponding Pass@1 entry, using the same donor and target checkpoint. Bold scores indicate the best result within each transfer direction, including ties.

Table 6: **Target-side adaptation after transfer.** MATH500 Pass@1 (%) at each target update.

Target	Method	0	10	25	50	75	100
Qwen2.5-1.5B	GrowRL	22.625	36.150	38.100	34.750	36.325	36.925
	LoRA	7.825	7.600	8.450	9.450	11.325	12.225
Qwen2.5-7B	GrowRL	40.875	56.175	52.125	54.550	53.575	55.925
	LoRA	34.775	35.725	36.750	39.125	40.650	41.500
Llama-3.2-3B	GrowRL	9.625	9.850	9.800	10.575	9.200	9.800
	LoRA	4.450	5.100	5.200	5.400	5.525	6.100

Table 7: **Immediate transfer performance.** Sampled Pass@8 (%) without target-side training.

Method	MATH	AMC	AIME	Minerva	Olymp.	GSM8K	HEval	STEM	ARC-C	MMLU-H
Qwen2.5-Math-1.5B → Qwen2.5-1.5B										
Base	27.80	20.00	0.00	5.88	7.72	36.39	19.51	40.92	45.22	39.28
LoRA	35.80	20.00	0.00	12.87	10.53	47.54	31.10	45.00	48.89	41.64
GrowRL	**65.20**	**40.00**	**6.67**	**20.22**	**26.41**	**80.29**	**48.78**	**53.51**	**56.74**	**43.49**
FineMath-Llama-3B → Llama-3.2-3B										
Base	20.00	17.50	**0.00**	12.13	6.97	23.88	33.54	47.32	54.44	56.17
LoRA	23.20	**22.50**	**0.00**	15.44	8.46	35.86	42.07	**52.05**	55.20	58.89
GrowRL	**30.60**	17.50	**0.00**	**18.75**	**9.64**	**55.34**	**48.78**	44.63	**63.14**	**63.93**
Qwen2.5-Math-7B → Qwen2.5-7B										
Base	76.80	57.50	**20.00**	34.93	36.80	95.68	78.05	89.86	96.84	84.57
LoRA	76.60	**62.50**	16.67	36.40	35.61	95.22	79.27	91.98	96.84	86.27
GrowRL	**80.00**	55.00	16.67	**41.91**	**37.98**	**96.06**	**87.20**	**92.05**	**97.44**	**88.20**

A.5 Trainable components and transfer strength

The component ablation in Table 5 compares *Full pathway*, which trains the complete added pathway, with W_O, W_d *only*, which trains only the added attention output projection W_O and FFN down projection W_d. In the latter variant, W_Q, W_K, W_V, W_g, W_u remain at their random initialization. The backbone, including the language-model output head, is frozen in both variants. Both variants contain an attention branch and an FFN branch; their trainable parameter counts are reported in the table. Bold scores indicate the better result within each backbone.

The transfer-strength experiment uses Qwen2.5-Math-1.5B $\rightarrow$ Qwen2.5-1.5B without target-side training. We vary the contribution of the transferred components with $\alpha \in \{0, 0.05, 0.1, 0.2, 0.5, 1\}$. As defined in Equation (5), GrowRL scales both branch outputs at every layer; LoRA scales each transferred adapter update. All learned parameters remain fixed during the sweep.

This analysis uses VAL300, a fixed set of 300 Level 3–5 problems sampled from the MATH test split, stratified by difficulty (92 Level 3, 99 Level 4, and 109 Level 5 problems). We verified that VAL300 has no overlap with the 5,585 RL training problems by either problem ID or normalized problem text. Each problem receives four sampled responses at temperature 1. The plotted mean binary reward is sampled Pass@1 on this validation set.

B Method and Implementation Details

B.1 Function preservation and initial gradients

At initialization, $W_O = W_d = 0$ makes both added residual corrections identically zero. Inducting over the attention and feed-forward sublayers in Equation (1) shows that every hidden state, and hence the policy distribution, equals that of the pretrained model in exact arithmetic. The same argument applies to inserting a fresh zero-output pathway into an adapted model. A transferred learned pathway instead uses nonzero output projections and can change the receiving model's policy immediately.

For either branch, write its output as $B_\psi(X)W_{\text{out}}$ and let D denote the loss gradient at that output. Then

$$\nabla_{W_{\text{out}}}\mathcal{L} = B_\psi(X)^\top D, \qquad \nabla_\psi \mathcal{L} = J_\psi[\operatorname{vec} B]^\top \operatorname{vec}\big(DW_{\text{out}}^\top\big). \tag{6}$$

Here $J_\psi[\operatorname{vec} B]$ is the Jacobian of the vectorized internal features with respect to the internal projection parameters. Thus, at zero output initialization, the internal projection gradients vanish, while the output gradients can be nonzero. Output changes allow internal learning on subsequent updates. The W_O, W_d-only control keeps the internal projection matrices at their random initialization; its inputs and activations can still change as earlier branches learn.

B.2 Complete RL objective

We train the pathway using Dr. GRPO (Liu et al., 2025). Let $\pi_{\theta,\phi}$ denote the policy with frozen backbone parameters θ and trainable pathway parameters ϕ. For each prompt x, we sample G completions $y_1, \ldots, y_G$ and obtain binary answer-verification rewards $R_i \in \{0, 1\}$. Each token in completion i receives the group-centered advantage

$$a_i = R_i - \frac{1}{G}\sum_{j=1}^{G} R_j, \qquad \rho_{i,t}(\phi) = \frac{\pi_{\theta,\phi}(y_{i,t} \mid x, y_{i,<t})}{\pi_{\text{old}}(y_{i,t} \mid x, y_{i,<t})}, \tag{7}$$

where π_{old} is the actor before the update. Following clipped policy optimization (Schulman et al., 2017), we minimize the following per-prompt loss, averaged over the prompt batch:

$$\mathcal{L}(\phi) = -\frac{1}{GT_{\max}}\sum_{i=1}^{G} w_i \sum_{t=1}^{|y_i|} \min\big[\rho_{i,t}(\phi)a_i,\ \operatorname{clip}\big(\rho_{i,t}(\phi), 1-\epsilon, 1+\epsilon\big)a_i\big]. \tag{8}$$

Here $T_{\max}$ is a fixed completion-length limit. We use $G = 8$, $T_{\max} = 3000$, and $\epsilon = 0.2$, without reward-standard-deviation normalization or a KL penalty. The fixed denominator avoids dividing each completion's contribution by its own length.

The detached weight w_i accounts for differences between the rollout engine and the training actor. If b denotes the rollout policy, the sequence importance ratio is $s_i = \pi_{\text{old}}(y_i \mid x)/b(y_i \mid x)$. Our GrowRL runs use $w_i = s_i \mathbf{1}[s_i \le 3]$; identical rollout and actor probabilities give $w_i = 1$. This sequence-level correction is separate from the token-level clipping in Equation (8). The same objective trains newly inserted pathways after transfer or composition, with only the newly added pathway parameters updated.

B.3 Implementation

For source-model RL, the added attention and FFN branches can be represented by concatenating their channels and output projections with those of the backbone. Zero-output padding heads preserve the backbone's grouped-query attention layout (Ainslie et al., 2023) where required, while each layer has one effective added attention head. Frozen backbone and padding slices are restored after optimizer steps. Sequential transfer retains the explicit residual steps and additional normalization layers in Equation (4).

Forward computation uses BF16, with FP32 master parameters and optimizer states. The parameter counts in the paper measure trainable pathway parameters; dense storage and optimizer allocation can be larger. Frozen backbone parameters remain unchanged, while the hidden states can change as the pathway learns.