Breaking Domain Barriers: Robust Prefix Optimization without Target Labels

Anonymous ACL submission

Abstract

Prompt Tuning (PT) has recently shown remarkable success in diverse Natural Language Processing (NLP) tasks, providing an efficient knowledge transfer paradigm to textually instruct models with domain-level guidance. However, PT approaches mix domain invariant reasoning with source artifacts, causing negative transfer on unseen domains and tasks. They rely on target labels or oracle checkpoint selection, while adversarial training and calibration seldom align training scaffolds, source only optimization, and uncertainty control, limiting deployment under distribution shift. Motivated by this, we propose **A**daptive **R**obust **P**refix **O**ptimization **(ARPO)**, a target-label-free tuning framework that combines disentangled prefix learning, training-consistent robustness, and Multi-Objective Bayesian Optimization (MOBO) for cross-domain transfer. Specifically, the disentangled prefix interface separates domain invariant reasoning from source-specific scaffolds through a shared interface, soft verbalizer, and auxiliary head. Training consistent robustness combines progressive DS removal, invariance regularization, and prefix first gating, making deployment rely on DI under strict zero shot transfer. Source-only optimization uses MOBO to balance gated accuracy, invariance, stability, and cost without target-label leakage. Extensive results on NLU datasets that show ARPO achieves 3% improvement in two experimental settings, highlighting its cross-domain generalization.

1 Introduction

Large language models (LLMs) have become the backbone of modern natural language understanding (Radford et al., 2019; Raffel et al., 2020; Brown et al., 2020; Touvron et al., 2023), while parameter-efficient fine-tuning, especially prefix tuning (Radford et al., 2021; Li and Liang, 2021) and soft prompt tuning (Lester et al., 2021; Liu et al., 2022b), adapts frozen backbones at low cost. Its promise is simple. A few trainable input vectors should let one frozen model serve many downstream domains (Houlsby et al., 2019; Hu et al., 2021; **?**; Vu et al., 2022). However, this promise breaks in cross-domain and cross-task settings. Prefix representations entangle transferable reasoning with source artifacts, making prompts hard to reuse (Bao, 2021; Dang et al., 2024; Bai et al., 2024b). Transferring such prefixes to unseen targets may even underperform the bare language model, causing negative transfer (Wang et al., 2023; Shi and Lipani, 2023; Bai et al., 2024b). Moreover, some methods use target labels or validation for checkpoint selection (Zhao et al., 2024; Bai et al., 2024a), and deployment shift can turn prefix predictions from helpful to harmful (Ganin et al., 2016; Madry et al., 2017).

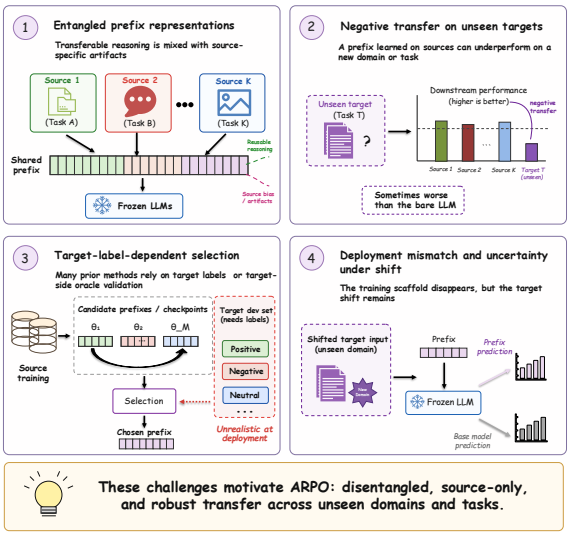

Figure 1: Four challenges in prefix transfer: entangled representations, negative transfer, target label selection, and deployment shift. They motivate ARPO for disentangled, source only, robust transfer.

Previous research in prefix tuning mainly reduces computational overhead by updating a small subset of parameters, including Adapters (Houlsby et al., 2019), LoRA (Hu et al., 2021), SPoT (Vu

et al., 2022), and P-Tuning v2 (Liu et al., 2022b). However, these methods still struggle with semantic and lexical shifts, making learned prefixes prone to negative transfer on unseen domains and tasks. Multi-part prompt methods such as ATTEMPT (Asai et al., 2022), MPT (Wang et al., 2023), and DePT (Shi and Lipani, 2023) improve modularity but lack systematic reuse across task families. Disentangled representation methods separate domain-specific and invariant features (Bao, 2021; Dang et al., 2024; Bai et al., 2024b), yet often keep source style active at inference. Adversarial training and calibration improve robustness (Ganin et al., 2016; Madry et al., 2017; Zhao et al., 2021), but rarely address deployment mismatch or source-only selection. Although multi-objective Bayesian optimization balances accuracy, robustness, and cost (Laumanns and Ocenasek, 2002; Daulton et al., 2020; Suzuki et al., 2020), it remains underexplored for target-label-free prefix optimization.

To this end, we propose ARPO, a target-label-free cross-domain prefix-tuning approach that integrates adaptive prefix disentanglement with training-consistent robustness and source-only optimization. Specifically, we construct a dynamically adaptive prefix interface by combining Information Bottleneck theory (Tishby et al., 2000) with orthogonality and adversarial invariance constraints, effectively separating reusable domain-invariant reasoning from source-specific style and annotation artifacts. Furthermore, we present a progressive source-scaffold removal strategy, a null-input calibrated confidence gate, and a source-only MOBO selection protocol to optimize accuracy, stability, invariance, and cost without target-label leakage, boosting robust cross-domain generalization.

Extensive experiments including GLUE (Wang et al., 2018), SuperGLUE (Wang et al., 2019), and MRQA 2019 (Fisch et al., 2019), together with robustness and ablation analyses, demonstrate the effectiveness of our approach. Our main contributions are summarized as follows:

(1) We propose a **disentangled prefix and interface design** that stores reusable reasoning in a domain-invariant prefix, absorbs source artifacts with a domain-specific branch, and reduces label-token bias through a verbalizer-head interface;

(2) We design a **training-consistent robustness mechanism** combining progressive domain-specific dropout, invariance-oriented regularization, null-input calibration, and prefix-first confidence gating against harmful prefix activation under shift;

(3) We introduce a **fully source-only selection and optimization protocol** that uses source proxy scoring, saturated-mode weighting, stochastic weight averaging, and MOBO to select stable configurations without target-label leakage.

2 Related Works

2.1 Domain Adaptation and Prefix Tuning.

Traditional domain adaptation usually updates all model parameters, which brings high computation and overfitting risks (Radford et al., 2021; Touvron et al., 2023; Han et al., 2024; Zaken et al., 2022). Parameter efficient methods reduce this cost through small trainable modules, including Adapters (Houlsby et al., 2019), LoRA (Hu et al., 2022a), and AdapterDrop (Rücklé et al., 2020), but they may still add latency or fail under large semantic shifts (Houlsby et al., 2019; Zhong et al., 2021). Prefix Tuning (Li and Liang, 2021), Prompt Tuning (Lester et al., 2021), SPoT (Vu et al., 2022), and P-Tuning v2 (Liu et al., 2022b) prepend continuous vectors to frozen models and further reduce trainable parameters, yet they remain sensitive to vocabulary, input, and domain perturbations (Ma et al., 2022). Recent multi part prompt methods, including ATTEMPT (Asai et al., 2022), MPT (Wang et al., 2023), and DePT (Shi and Lipani, 2023), improve modularity and transferability but still lack systematic reuse across unseen domains and tasks. Differently, our solution builds a unified invariant and specific prefix structure to separate reusable reasoning from source artifacts.

2.2 Adversarial Training and Multi-Domain Robustness.

Adversarial training improves robustness in NLP through perturbation based objectives (Miyato et al., 2018), but embedding perturbations can mismatch discrete text changes. Token aware training narrows this gap (Li and Qiu, 2021) but adds complexity. Contrastive, curriculum, adversarial distributional, and adversarial self training improve shift generalization (Rim et al., 2021; Yoo and Qi, 2021; Dong et al., 2020; Shi and Liu, 2023). Disentangled prompt and information constrained methods separate signal from noise (Bai et al., 2024b; Zhao et al., 2024; Bai et al., 2024a; Alemi et al., 2016; Wu et al., 2023). Yet source style may remain active, so prefixes fail on unfamiliar inputs. ARPO combines adversarial robustness, domain specific prefix dropout, and confidence gated decoding for

cross-domain robustness with minimal overhead.

2.3 Source-Only Selection and Multi-Objective Bayesian Optimization.

Reliable cross-domain evaluation requires avoiding target side leakage in selection. Strong oracle protocols such as SPoT, MPT, and DePT tune checkpoints or thresholds on target holdouts, weakening zero shot claims (Vu et al., 2022; Wang et al., 2023; Shi and Lipani, 2023); weaker protocols use unlabeled target data and sit between zero shot and semi supervised evaluation (Chen et al., 2023; Phan et al., 2024). Verbalizers may also inject bias through external knowledge, task prototypes, or context free label priors (Hu et al., 2022b; Cui et al., 2022; Zhao et al., 2021). Bayesian Optimization offers sample efficient search for costly objectives (Snoek et al., 2012; Shahriari et al., 2015; Frazier, 2018), while scalable multi objective variants use Gaussian processes, trust regions, and hypervolume improvement to balance goals (Jin et al., 2018; Eriksson et al., 2019; Emmerich et al., 2011; Daulton et al., 2020; Ru et al., 2020). Yet existing MOBO mainly serves generic hyperparameter search, not source-only prefix optimization. ARPO optimizes accuracy, robustness, stability, and cost through source proxy scoring, SWA, and MOBO without target-label leakage.

3 Methodology

We introduce ARPO in Figure 2, a target-label-free framework for cross-domain transfer with frozen decoder LMs. The method follows three mechanisms: (i) a DI/DS prefix decomposition that uses source-specific capacity as a training scaffold and deploys only the invariant branch (Sec 3.1), (ii) a prefix-first deployment gate with null calibration for confidence under shift (Sec 3.2), and (iii) a source-only selector and MOBO search protocol that optimizes accuracy, invariance, stability, and cost without target labels (Sec 3.3). Objective terms act as stabilizers rather than separate contributions; they regularize routing, auxiliary prediction, prompt perturbation, consistency, residual scale, and source-specific averaging. Their role is to keep the three mechanisms trainable under source-only supervision, while Sec 4.3 focuses on the architectural and selection components. Sec 3.4 describes how these mechanisms form a single loop across training, selection, and deployment. Appendix A gives module sizes, loss weights, default settings, and the set of components used in the main results.

3.1 Disentangled Prefix Interface

The backbone f_θ stays frozen throughout training. We update the prefix, router, discriminators, verbalizer weights, and output modules. Following prefix tuning and prompt tuning (Li and Liang, 2021; Lester et al., 2021), each input is prepended with a prefix and split into two branches:

$$\begin{aligned} P(x,d) &= \big[P_{\mathrm{DI}}(x); P_{\mathrm{DS}}^{(d)}(x)\big], \\ P_{\mathrm{DI}} &\in \mathbb{R}^{L_{\mathrm{DI}}\times h}, \quad P_{\mathrm{DS}} \in \mathbb{R}^{L_{\mathrm{DS}}\times h}. \end{aligned} \tag{1}$$

We use $L_{\mathrm{DI}} = 25$, $L_{\mathrm{DS}} = 25$, and $L = 50$ in the main setting. The DI branch is shared across all source domains and carries the signal used at deployment. The DS branch is indexed by source domain d and used only as a training scaffold; it is removed at target evaluation and never serves as a target-domain adapter.

Let $\bar{e}(x)$ denote the masked mean input embedding. A lightweight router maps $\bar{e}(x)$ to a soft mixture over $R = T + 1$ task type experts, with temperature $\tau = 1.5$:

$$\pi(x) = \mathrm{softmax}\big(g_\rho(\bar{e}(x))/\tau\big). \tag{2}$$

Here T is the number of source *task types*, such as NLI, sentiment, paraphrase, and Boolean QA, and is distinct from source domains d. Experts operate at task type level rather than domain level, so router is not used as a domain classifier. Additional expert in $R = T + 1$ is shared across task types.

The DI prefix combines a global base, routed expert residuals, an input-conditioned gate, and a small residual correction:

$$\begin{aligned} M_{\mathrm{DI}}(x) &= B_0 + \sum_{r=1}^{R} \pi_r(x)\, B_r, \\ P_{\mathrm{DI}}(x) &= g_\phi(\bar{e}(x)) \odot M_{\mathrm{DI}}(x) + P_{\mathrm{res}}(x). \end{aligned} \tag{3}$$

The gate $g_\phi(\bar{e}(x)) \in (0,1)^{L_{\mathrm{DI}}}$ controls how much of the shared mixture is exposed at each prefix position. The bottlenecked residual term gives limited input-conditioned correction, while the routed shared prefix remains the main carrier of transferable signal. The DS branch is kept lightweight:

$$P_{\mathrm{DS}}^{(d)}(x) = \bar{P}_{\mathrm{DS}}^{(d)} + P_{\mathrm{DS,res}}^{(d)}(x). \tag{4}$$

Each source domain has a trainable scaffold with an instance-conditioned adjustment. This branch

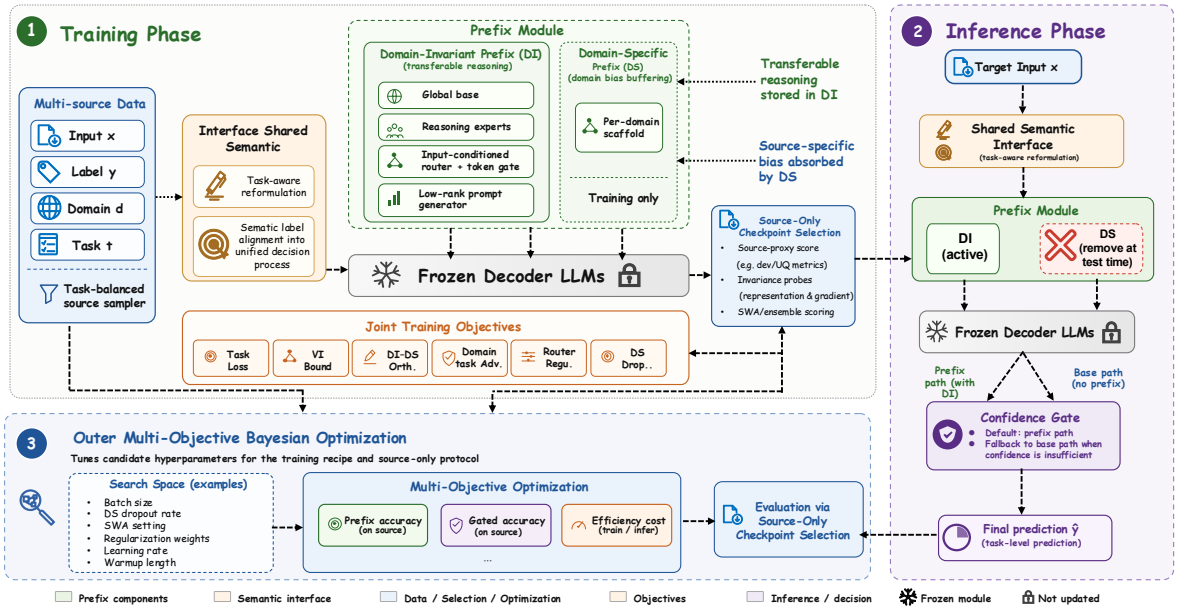

Figure 2: Overall framework of ARPO. A frozen decoder LM is adapted through a compact prefix interface. The DI branch carries the deployed decision signal, while the DS branch provides source-side scaffolding during training and is removed at inference. A shared verbalizer, auxiliary head, prefix-first gate, and source-only MOBO selection rule support transfer without target labels.

gives variation a training outlet, so DI is not rewarded for storing information that disappears once DS is removed. At target evaluation, unseen task ids fall back to the shared path rather than being mapped to a source-specific head.

The output side is shared across tasks. A template rewrites x as a question for task type t, and Ψ_t projects the native label space $\mathcal{Y}_t$ into a common answer set $\mathcal{A} = \{\text{yes}, \text{no}\}$, with $C = 2$. This shared interface defines the projected decision space used throughout cross-task evaluation and controls label-space mismatch across source families without adding target-specific shortcuts. For tasks whose native labels are not binary, the mapping procedure is specified in Appendix A. Each answer class owns a small candidate pool $\mathcal{C}^{(c)}$. The verbalizer and auxiliary classifier are then combined in probability space:

$$\begin{aligned} \omega_{c,k} &= \mathrm{softmax}_k(w^{(c)}/\tau_w), \\ \ell_c^{\mathrm{verb}}(x) &= \sum_{k=1}^{K_c} \omega_{c,k}\, \ell_{a^{(c,k)}}(x), \\ p_{\mathcal{A}}(c \mid x) &= \alpha\, s_c^{\mathrm{verb}}(x) + (1-\alpha)\, s_{t,c}^{\mathrm{aux}}(x). \end{aligned} \tag{5}$$

We set $\alpha = 0.85$. The auxiliary head reads a detached hidden state, allowing it to stabilize answer-token probabilities without turning the shared prefix into a source-specific classifier. If the task id is missing or outside the source set, the shared auxiliary head is used.

The DI/DS split is a testable hypothesis rather than a naming convention. We use an information-theoretic view without claiming disentanglement: DI should keep $I(h_{\mathrm{DI}}; Y)$ high while reducing $I(h_{\mathrm{DI}}; D)$ and $I(h_{\mathrm{DI}}; T)$, and DS should absorb source-specific variation associated with $I(h_{\mathrm{DS}}; D)$ that should not be used at test time. Adversarial loss, orthogonality penalty, VIB term, and progressive DS dropout introduced in Sec 3.2 act as tractable surrogates for these quantities. This view makes design falsifiable: DI should remain sufficient at inference, DS should not transfer reliably, and the DI/DS length ratio should have an interior optimum. Appendix C.3 formalizes this training–deployment decomposition, while Sec 4.3 and Table 5, 4 provide the empirical evidence for the DI-only inference and length-ratio predictions.

3.2 Training Objective and Prefix-First Inference

Both branches are active during training, but inference uses only DI. To expose this mismatch, we progressively weaken DS:

$$\begin{aligned} \tilde{p}(e) &= p_0(0.25 + 0.75 p_e), \\ m_{\mathrm{DS}} &\sim \mathrm{Bernoulli}(1 - \tilde{p}(e)), \\ P_{\mathrm{train}}(x, d) &= \left[P_{\mathrm{DI}}(x); m_{\mathrm{DS}}\, P_{\mathrm{DS}}^{(d)}(x)\right]. \end{aligned} \tag{6}$$

Here $p_e \in [0, 1]$ is training progress and $p_0 = 0.15$. The schedule gives source-specific variation early access to DS, then gradually makes DI responsible for the signal available at deployment; Appendix C.5 formalizes this interpolation. We also inject prompt-space Gaussian noise into P_{DI}, using the default scale in Appendix A, without modifying the frozen backbone. The supervised loss is computed on the shared answer set:

$$\mathcal{L} = \mathrm{CE}\big(p_{\mathcal{A}}(\cdot \mid x), \Psi_t(y)\big) + \sum_{i \in \mathcal{S}} \lambda_i \mathcal{L}_i. \tag{7}$$

$\mathcal{S}$ implements the DI/DS hypothesis through invariance, separation, routing, and stability terms. VIB and DI–DS separation shape the representation; gradient-reversal adversaries (Ganin et al., 2016) reduce recoverable domain and task traces in DI; router terms prevent expert collapse; the remaining terms control prediction balance, residual scale, consistency, and the auxiliary interface. Default weights are given in Appendix A; Sec 4.3 reports ablations for the main components used in the analysis. PGD, when enabled, follows the detached-difference form in Appendix A.

At inference, DS is zeroed and the model evaluates two paths. The prefix path uses P_{DI} with the interface in Eq. 5; the base path drops all prefixes and reads the frozen LM over $\mathcal{A}$. At test time, $P_{\mathrm{test}}(x) = \big[P_{\mathrm{DI}}(x); 0\big]$. Null-input biases are estimated once per evaluation run, giving calibrated logits $\ell^g_{\mathrm{pref}} = \ell_{\mathrm{pref}} - \ell^{\mathrm{null}}_{\mathrm{pref}}$ and $\ell^g_{\mathrm{base}} = \ell_{\mathrm{base}} - \ell^{\mathrm{null}}_{\mathrm{base}}$ for confidence and entropy. Class identity is read from the raw logits of each path, so calibration affects only confidence and entropy.

The gate is prefix-first. Let $\hat{y}^r, q^r, H^r$ denote prediction, confidence, and entropy on path $r \in \{\mathrm{pref}, \mathrm{base}\}$. The prefix prediction is accepted when the two paths agree, or when the prefix path is more certain than the base path and clears the source-fixed confidence floor:

$$\begin{aligned}\mathcal{B}(x) = \mathbf{1}\{\hat{y}^{\mathrm{pref}} = \hat{y}^{\mathrm{base}}\} \\ \vee \Big[\mathbf{1}\{H^{\mathrm{base}} - H^{\mathrm{pref}} \geq \theta_H\} \\ \cdot \mathbf{1}\{q^{\mathrm{pref}} \geq \theta_{\min}\}\Big].\end{aligned} \tag{8}$$

Thresholds are fixed by the source-only protocol formalized in Appendix C.2 and reported in Appendix A. The decoded label is

$$\hat{y}(x) = \begin{cases} \hat{y}^{\mathrm{pref}}(x), & \mathcal{B}(x) = 1, \\ \hat{y}^{\mathrm{base}}(x), & \mathcal{B}(x) = 0. \end{cases} \tag{9}$$

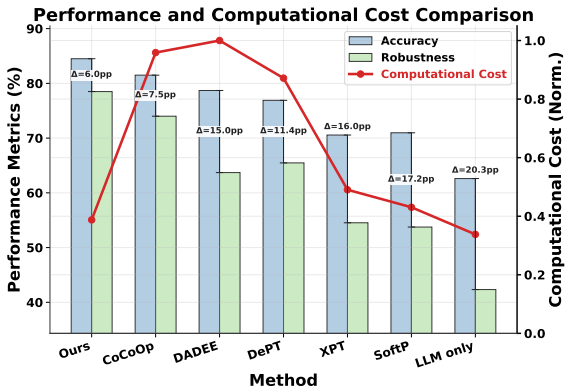

Figure 3: Accuracy, robustness, and normalized cost on Qwen3.5 9B with an A6000 GPU.

The acceptance rule keeps the prefix as the default path, while allowing the bare LM to act as a fallback under source-fixed thresholds. These thresholds are chosen once on a held-out source-side validation split by maximizing the source proxy score in Eq. 11, then frozen across all transfer pairs, backbones, and reported tables. No target-side signal is used. Appendix C.6 gives the risk-decomposition analysis of this source-fixed gate.

Checkpoint selection follows the no-target-label rule. The sampler draws a task type uniformly, then a source domain within that task type, and finally a minibatch. At the end of each epoch, ARPO evaluates a *source proxy set*, a task-balanced held-out split from training sources shared by every method. Two linear probes estimate whether source domain and task type remain recoverable from h_{DI}. For $z \in \{D, T\}$, let a_z be probe accuracy and $b_z = 1/N_z$ be the chance baseline:

$$u_z = \mathrm{clip}_{[0,1]}\left(1 - \frac{a_z - b_z}{1 - b_z}\right), \quad z \in \{D, T\}. \tag{10}$$

The source proxy score then combines gated source accuracy, uncertainty, and the two invariance probes:

$$\begin{aligned}\mathrm{Score} = w_s a^{\mathrm{gated}}_{\mathrm{src}} + w_e \bar{H}_{\mathrm{pred}} \\ + w_D u_D + w_T u_T.\end{aligned} \tag{11}$$

Here $a^{\mathrm{gated}}_{\mathrm{src}}$ measures deployment-consistent source accuracy, $\bar{H}_{\mathrm{pred}}$ measures predictive uncertainty, and u_D, u_T penalize recoverable domain and task traces in DI. After source accuracy saturates, the selector shifts weight toward uncertainty and invariance, then applies late-stage SWA only when the source proxy metric is preserved. Appendix A lists the weights, saturation cutoff, SWA window, and acceptance tolerance; Appendix C.7

proves that the score is source-admissible under the no-target-label protocol.

3.3 Source-Only Outer Search

ARPO separates model design from source-side selection, with target labels excluded from both. The outer search follows the checkpoint-selection constraint: target accuracy is unobservable, and saturated source accuracy becomes weak. MOBO fits this multi-objective target by balancing gated source accuracy, invariance, late-stage stability, and cost under the source-only protocol. Each candidate uses the same sampler, the selector in Eq. 11, and the gate in Eq. 8. Thus, the search changes the training recipe, not the evaluation rule. The configuration in Appendix A is selected through this source-proxy process, so results in Sec 4 are not tuned with target labels.

The search vector includes the learning rate, adversarial warmup, and main coefficient groups in Eq. 7: invariance, separation, routing, consistency, residual regularization, and auxiliary terms. Other components are activated only for ablations. The MOBO loop evaluates four source side objectives:

$$\begin{aligned} J_{\text{acc}}(\omega) &= 1 - a_{\text{src}}^{\text{gated}}(\omega), \\ J_{\text{inv}}(\omega) &= 1 - \tfrac{u_D(\omega)+u_T(\omega)}{2}, \\ J_{\text{stab}}(\omega) &= \text{Var}_{e\in\mathcal{E}_{\text{late}}}(\text{Score}_e(\omega)), \\ J_{\text{cost}}(\omega) &= \log(1+\text{time}(\omega)). \end{aligned} \tag{12}$$

These objectives make MOBO fit the selection problem by preserving gated accuracy, reducing recoverable domain and task traces in DI, maintaining late training stability, and controlling cost. A Sobol design initializes the search, then a surrogate proposes candidates by Pareto improvement. Only source-proxy Pareto-front candidates receive the full training budget, and the final configuration is selected by Eq. 11 under the same saturation rule. Thus, the outer loop cannot act as a hidden target selector. Appendix C.8 formalizes the source-only admissibility of the MOBO loop, while Appendix A gives the corresponding budget and implementation settings.

3.4 How Three Mechanisms Work Together

The three core mechanisms share one design axis: training-time signals differ from deployment-time signals. Sec 3.1 addresses this at the representation level: DI and DS coexist during training, but only DI remains active at inference, so DI must keep the shift-robust signal. Sec 3.2 addresses the decision level: since prefix reliability is unverifiable under shift, null calibration and the prefix-first gate decide whether to trust DI or fall back to the bare LM. Sec 3.3 addresses selection: source proxy signals are observable but target labels are unavailable, so the selector in Eq. 11 and MOBO objectives in Eq. 12 remain source-side.

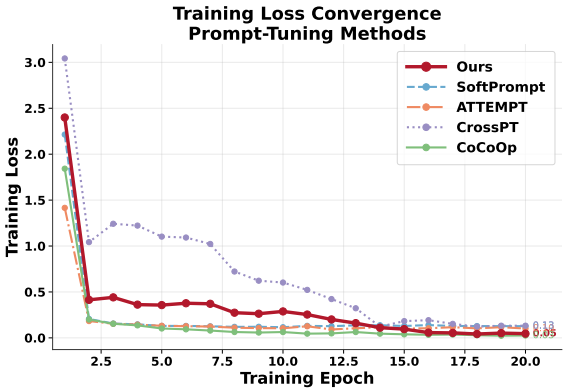

Figure 4: Training loss convergence across methods.

These three layers form a coupled loop rather than independent add-ons. The DI/DS split makes invariance probes u_D, u_T meaningful; progressive DS dropout makes the selector depend on a DI branch trained with reduced DS support; and the prefix-first gate turns $a_{\text{src}}^{\text{gated}}$ into a deployment-consistent signal rather than raw prefix accuracy. MOBO searches the same source-side quantities, so the outer loop adds no target-facing criterion. Sec 4.3 and Tables 5, 4 support this coupling: removing components weakens transfer, and balanced DI/DS gives the strongest configuration.

4 Experiments

We evaluate ARPO under a strict no target label protocol, where target labels are never used for training, tuning, or checkpoint selection. Experiments cover 12 NLU datasets from six families, including NLI, sentiment analysis, paraphrase detection, topic classification, Boolean QA, and acceptability judgment, all mapped to the shared discriminative interface in Sec 3.1. Datasets and training settings are in Appendix A, and extended experiments in Appendix B. We consider CombA for within family domain and style transfer and CombB for cross task generalization, using frozen Qwen3.5 9B, Qwen3.5 35B (Team, 2026), and Devstral2 24B (Rastogi et al., 2025) backbones under 4 bit quantization with only prefixes and auxiliaries updated. We compare with standard PEFT methods (Houlsby et al., 2019; Hu et al., 2022a; Li and Liang, 2021; Lester et al., 2021; Liu et al., 2022b), multi task

Table 1: CombA within-family transfer accuracy (%) across five held-out pairs, evaluating domain and style transfer under fixed reasoning types. Using 9B, 24B, and 35B. Mean averages five pairs. **Bold**: best; underline: second-best.

Method	RTE→SciTail			Yelp→SST 2			News→HP			MNLI→QQP			CoLA→PAWS			Mean		
	9B	24B	35B	9B	24B	35B	9B	24B	35B	9B	24B	35B	9B	24B	35B	9B	24B	35B
LLM only	63.9	67.8	70.2	73.7	74.5	79.2	40.4	43.7	45.1	67.7	69.1	72.3	60.5	61.2	66.7	61.2	63.3	66.7
CrossPT	70.4	72.1	75.6	80.9	82.3	85.7	69.2	70.3	74.2	67.8	69.2	71.8	63.4	64.8	67.9	70.3	71.7	75.0
Soft prompt	70.9	73.2	76.3	87.3	90.1	93.2	70.1	71.1	75.5	75.1	78.3	80.2	68.6	69.7	73.1	74.4	76.5	79.7
ATTEMPT	84.7	85.1	88.2	88.6	89.3	94.1	71.2	72.4	75.6	76.8	78.1	80.3	70.6	71.2	75.8	78.4	79.2	82.8
XPROMPT	75.2	76.1	78.2	90.3	90.6	94.3	75.4	76.2	79.1	84.4	85.3	89.3	73.8	74.5	76.8	79.8	80.5	83.5
InfoPrompt	76.7	77.1	80.2	91.5	92.4	95.5	74.1	75.2	79.8	85.8	86.8	90.5	75.2	76.6	79.1	80.7	81.6	85.0
UAdapter	77.9	79.2	82.9	91.6	92.8	96.7	74.8	75.5	79.5	87.2	88.1	91.7	75.4	76.8	79.3	81.4	82.5	86.0
DePT	77.4	79.2	81.3	90.1	92.3	95.5	73.5	74.2	78.2	85.2	87.1	91.3	75.1	75.9	79.6	80.3	81.7	85.2
DADEE	78.5	79.6	82.8	91.9	92.7	96.8	75.1	76.7	79.6	87.8	89.1	93.3	76.1	77.9	80.1	81.9	83.2	86.5
CoCoOp	88.1	88.9	93.2	87.4	88.7	92.9	79.4	80.5	84.6	76.1	77.2	79.8	75.1	76.2	79.5	81.2	82.3	86.0
ARPO	**91.3**	**92.1**	**95.8**	**92.7**	**93.5**	**96.9**	**82.6**	**83.3**	**86.2**	**88.3**	**89.2**	**93.5**	**77.2**	**78.4**	**81.1**	**86.4**	**87.3**	**90.7**

prefix methods (Vu et al., 2022; Asai et al., 2022; Wang et al., 2023; Shi and Lipani, 2023; Liu et al., 2024; Pouramini and Faili, 2025), and adaptive or verbalizer based methods (Jain et al., 2024; Tang et al., 2025; Bai et al., 2024b; Zhou et al., 2022; Li et al., 2024; Das et al., 2024; Liu et al., 2022a; Cui et al., 2022; Hu et al., 2022b), while keeping the same interface, mixing rule, and source proxy hyperparameter selection for all methods.

4.1 Main Results

Table 1, 2 report the main transfer results. ARPO obtains the best mean accuracy on all three backbones, with 86.4, 87.3, and 90.7 on CombA, and 83.1, 84.4, and 88.3 on CombB. On CombA, it improves over the strongest baseline DADEE by 4.5, 4.1, and 4.2 points, and exceeds LLM only by 25.2, 24.0, and 24.0 points. On CombB, it improves over the strongest baseline CoCoOp by 1.4, 1.8, and 2.2 points, and exceeds LLM only by 18.8, 19.4, and 19.9 points. Appendix B.1 confirms run stability. The harder cross task setting exposes negative transfer in some baselines, where InfoPrompt and UAdapter fall below LLM only on NSQ→Para, while ARPO is stable and increases from 9B to 35B by 4.3 points on CombA and 5.2 points on CombB.

The results support the motivation of ARPO. The DI branch preserves transferable reasoning, while the DS branch absorbs source style and is removed at inference, reducing source artifact influence on unseen targets. The router, soft verbalizer, and prefix-first gate stabilize prediction under task mismatch, keeping gains across within family and harder cross task transfer without target labels.

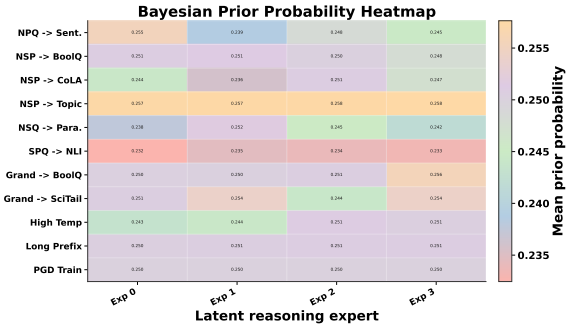

Figure 5: Bayesian prior probability heatmap showing balanced expert usage / no collapse.

4.2 Robustness Analysis

Figures 3, 4, and Table 3, show that ARPO is most robust under token, phrase, and swap perturbations, with the highest mean robust accuracy of 78.4 and the smallest robustness gap of 6.0 points. It improves over CoCoOp by 5.0 points, stays above 76.1 across perturbations, and avoids the drops near 60 seen in DePT and DADEE on CombB cases. Although CoCoOp reaches lower source loss than ARPO at 0.03 versus 0.05, its larger robustness gap shows that source convergence alone cannot ensure stable transfer. This matches ARPO, where prompt noise, pair consistency, attention concentration, and prefix-first gating reduce token, swap, answer, and shift sensitivity.

Figure 5 shows balanced Bayesian priors around 0.24 to 0.26, with uniform PGD Train and Long Prefix and only mild NSP→Topic preference. Entropy patterns further indicate that multi source

Table 2: CombB cross-task-family transfer accuracy (%) across six held-out pairs, evaluating reusable reasoning across different task families. Using 9B, 24B, and 35B. Mean averages six pairs. **Bold**: best; underline: second-best.

Method	NSP→BoolQ			NSP→CoLA			NSP→Topic			SPQ→NLI			NPQ→Sent			NSQ→Para			Mean		
	9B	24B	35B	9B	24B	35B	9B	24B	35B	9B	24B	35B	9B	24B	35B	9B	24B	35B	9B	24B	35B
LLM only	59.9	60.2	63.4	69.5	70.1	74.2	58.4	60.1	62.2	63.9	64.6	67.8	64.3	65.1	68.3	69.5	70.1	74.6	64.3	65.0	68.4
Soft prompt	68.4	69.1	72.8	70.3	71.5	74.6	69.8	70.8	73.2	77.6	78.3	81.2	72.3	73.7	77.1	71.0	72.8	76.4	71.6	72.7	75.9
CrossPT	68.5	70.0	73.2	69.4	71.2	73.8	70.2	71.9	74.2	70.4	71.5	73.6	70.7	72.0	74.8	73.4	75.1	77.8	70.4	72.0	74.6
ATTEMPT	79.2	80.9	83.8	77.2	79.1	82.4	80.2	81.9	84.1	85.1	86.2	89.4	82.5	83.7	86.2	79.2	80.1	84.3	80.6	82.0	85.0
XPROMPT	69.1	70.2	73.5	78.2	79.3	82.6	73.5	74.1	75.8	76.6	77.5	79.3	78.2	79.4	81.7	73.5	74.1	78.6	74.9	75.8	78.6
InfoPrompt	69.6	70.4	73.5	71.0	72.3	76.5	73.9	74.6	76.5	77.0	78.2	80.1	78.8	79.5	81.7	64.1	65.4	68.3	72.4	73.4	76.1
DePT	70.5	71.8	74.3	72.5	73.7	76.2	73.0	73.9	77.1	77.6	78.5	81.8	79.1	79.8	83.7	68.5	69.8	72.2	73.5	74.6	77.6
UAdapter	72.4	73.0	76.1	74.2	75.3	78.8	74.6	75.5	78.9	81.5	82.3	85.4	76.0	76.8	79.9	65.8	66.3	70.0	74.1	74.9	78.2
DADEE	73.1	73.9	75.7	75.3	76.1	80.6	75.2	75.8	79.5	82.1	83.6	86.8	76.6	77.2	80.7	70.5	71.3	75.3	75.5	76.3	79.8
CoCoOp	84.3	85.1	88.5	76.6	77.4	80.3	78.8	79.2	83.1	88.7	89.4	92.6	83.5	84.7	89.2	78.4	79.5	83.0	81.7	82.6	86.1
ARPO	**85.2**	**86.7**	**90.1**	**79.7**	**80.6**	**85.2**	**80.8**	**82.1**	**85.6**	**89.2**	**90.5**	**94.3**	**84.2**	**85.9**	**90.1**	**79.6**	**80.4**	**84.7**	**83.1**	**84.4**	**88.3**

Table 3: Robustness under Token, Phrase, and Swap perturbations on 9B. ± denote standard deviations.

Method	Token		Phrase		Swap	
	A	B	A	B	A	B
LLM only	41.9±0.9	43.1±0.8	41.4±1.0	43.5±0.9	39.8±1.1	45.1±0.8
DePT	67.6±0.7	63.2±0.7	69.8±0.6	62.1±0.8	68.2±0.7	61.8±0.8
DADEE	66.4±0.8	61.8±0.8	64.3±0.9	58.2±0.9	67.4±0.7	59.9±0.8
CoCoOp	73.8±0.5	72.1±0.5	74.5±0.5	73.5±0.5	72.1±0.6	74.3±0.5
ARPO	**80.2**±0.4	**76.3**±0.4	**79.4**±0.5	**76.1**±0.4	**80.9**±0.3	**77.6**±0.4

Table 4: DI:DS length ablation at $L = 50$ over CombA and CombB in 9B and 35B model.

DI:DS	CombA		CombB	
	9B	35B	9B	35B
9:1	77.5±0.7	79.4±0.6	72.3±0.8	76.1±0.7
8:2	80.3±0.6	82.6±0.5	75.7±0.7	77.5±0.6
7:3	81.7±0.5	85.5±0.5	78.9±0.6	79.3±0.5
6:4	84.2±0.5	88.3±0.4	80.2±0.5	82.6±0.5
5:5	**86.4**±0.4	**90.7**±0.3	**83.1**±0.4	**88.3**±0.4
4:6	84.7±0.5	87.2±0.4	79.8±0.6	82.3±0.5
3:7	82.5±0.6	86.1±0.5	77.3±0.7	80.2±0.6
2:8	80.9±0.7	83.9±0.6	75.1±0.8	77.2±0.7
1:9	76.2±0.8	78.1±0.6	71.6±0.9	75.8±0.8

transfer needs mixed experts, supporting task related soft routing without expert collapse.

4.3 Ablation Study

Table 5, 4 show that both core modules and the DI DS budget are important for stable transfer, as the full model performs best on CombA and CombB with both 9B and 35B backbones. Removing Adv leads to the largest decline, from 86.4 to 79.8 on CombA 9B and from 88.3 to 81.3 on CombB 35B, while removing Bayesian, VIB, Reason, or Gate lowers performance by about 4.2 to 6.5 points. The 5:5 DI DS split also improves over 9:1 by 8.9, 11.3, 10.8, and 12.2 points across the four columns. These results are consistent with the design of ARPO, since Adv and VIB reduce source traces in DI, the Bayesian router preserves task related expert mixtures, and the gate limits unreliable prefix predictions under shift. The best 5:5 split further shows that DS must absorb source style during training, while DI must keep enough reasoning capacity for inference after DS is removed.

Table 5: Ablations Study on Qwen3.5 9B and 35B for CombA and CombB. Values are percentages; ± terms denote standard deviations.

Method	CombA		CombB	
	9B	35B	9B	35B
w/o Reason	81.3±0.5	84.2±0.3	78.2±0.3	83.5±0.2
w/o Bayes Router	80.5±0.6	85.4±0.3	77.8±0.5	82.8±0.5
w/o VIB	81.1±0.5	84.8±0.1	77.3±0.4	82.2±0.4
w/o Adv	79.8±0.7	83.2±0.4	76.2±0.6	81.3±0.6
w/o Gate	82.2±0.4	85.5±0.2	78.5±0.3	83.2±0.3
Full	**86.4**±0.4	**90.7**±0.3	**83.1**±0.3	**88.3**±0.5

5 Conclusion

To summarize, we propose ARPO, a target-label-free prefix tuning framework unifying prefix disentanglement, training consistent confidence gating, and source only MOBO for robust cross domain transfer. We separate P_{DI} and P_{DS} with routing and invariant constraints, progressively weaken source specific branch, and combine null calibrated prefix first gating, source proxy scoring, and late stage SWA to improve robustness stability tradeoff.

6 Limitations

ARPO is designed for target label free transfer, and its contribution is to couple DI and DS prefix disentanglement, null calibrated prefix first gating, and source only MOBO so that training, decoding, and checkpoint selection remain independent of target labels. This evaluation follows this design and covers diverse NLU benchmarks under within family and cross task transfer. However, the tasks are discriminative and are reformulated into a shared answer interface. This setting allows controlled evaluation of transferable reasoning, but it leaves open how the disentanglement and gating principles behave in open ended generation, structured prediction, native multi class classification, and multilingual transfer, where the output space is larger and the answer interface is less uniform.

A second limitation lies in the source side control signals used by ARPO. Source proxy validation, task templates, confidence thresholds, and MOBO objectives are introduced to avoid target label leakage and align model selection with deployment, but they may require careful implementation when source domains are highly imbalanced, weakly related, or noisy. In such cases, proxy accuracy, invariance probes, and calibration statistics may become less reliable, and the adversarial and routing modules may need additional diagnostics or adaptive weighting. Future work will extend ARPO to generative and cross lingual settings, improve automatic template construction and threshold calibration, and study more robust source proxy criteria for heterogeneous source mixtures.

References

Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. 2016. Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410*.

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. 2019. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*.

Akari Asai, Mohammadreza Salehi, Matthew E Peters, and Hannaneh Hajishirzi. 2022. Attempt: Parameter-efficient multi-task tuning via attentional mixtures of soft prompts. In *Proceedings of the 2022 conference on empirical methods in natural language processing*, pages 6655–6672.

Shuanghao Bai, Yuedi Zhang, Wanqi Zhou, Zhirong Luan, and Badong Chen. 2024a. Soft prompt generation for domain generalization. In *European Conference on Computer Vision*, pages 434–450. Springer.

Sikai Bai, Jie Zhang, Song Guo, Shuaicheng Li, Jingcai Guo, Jun Hou, Tao Han, and Xiaocheng Lu. 2024b. Diprompt: Disentangled prompt tuning for multiple latent domain generalization in federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 27284–27293.

Feng Bao. 2021. Disentangled variational information bottleneck for multiview representation learning. In *Artificial Intelligence: First CAAI International Conference, CICAI 2021, Hangzhou, China, June 5–6, 2021, Proceedings, Part II 1*, pages 91–102. Springer.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Haoran Chen, Xintong Han, Zuxuan Wu, and Yu-Gang Jiang. 2023. Multi-prompt alignment for multi-source unsupervised domain adaptation. *Advances in Neural Information Processing Systems*, 36:74127–74139.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. Boolq: Exploring the surprising difficulty of natural yes/no questions. *arXiv preprint arXiv:1905.10044*.

Ganqu Cui, Shengding Hu, Ning Ding, Longtao Huang, and Zhiyuan Liu. 2022. Prototypical verbalizer for prompt-based few-shot tuning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7014–7024.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The pascal recognising textual entailment challenge. In *Machine learning challenges workshop*, pages 177–190. Springer.

Zhuohang Dang, Minnan Luo, Chengyou Jia, Guang Dai, Jihong Wang, Xiaojun Chang, and Jingdong Wang. 2024. Disentangled representation learning with transmitted information bottleneck. *IEEE Transactions on Circuits and Systems for Video Technology*.

Sarkar Snigdha Sarathi Das, Ryo Kamoi, Bo Pang, Yusen Zhang, Caiming Xiong, and Rui Zhang. 2024. Greater: Gradients over reasoning makes smaller language models strong prompt optimizers. *arXiv preprint arXiv:2412.09722*.

Samuel Daulton, Maximilian Balandat, and Eytan Bakshy. 2020. Differentiable expected hypervolume improvement for parallel multi-objective bayesian optimization. *Advances in Neural Information Processing Systems*, 33:9851–9864.

Yinpeng Dong, Zhijie Deng, Tianyu Pang, Jun Zhu, and Hang Su. 2020. Adversarial distributional training

for robust deep learning. *Advances in Neural Information Processing Systems*, 33:8270–8283.

Michael TM Emmerich, André H Deutz, and Jan Willem Klinkenberg. 2011. Hypervolume-based expected improvement: Monotonicity properties and exact computation. In *2011 IEEE Congress of Evolutionary Computation (CEC)*, pages 2147–2154. IEEE.

David Eriksson, Michael Pearce, Jacob Gardner, Ryan D Turner, and Matthias Poloczek. 2019. Scalable global optimization via local bayesian optimization. *Advances in neural information processing systems*, 32.

Adam Fisch, Alon Talmor, Robin Jia, Minjoon Seo, Eunsol Choi, and Danqi Chen. 2019. Mrqa 2019 shared task: Evaluating generalization in reading comprehension. In *Proceedings of the 2nd workshop on machine reading for question answering*, pages 1–13.

Peter I Frazier. 2018. A tutorial on bayesian optimization. *arXiv preprint arXiv:1807.02811*.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario March, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *Journal of machine learning research*, 17(59):1–35.

Yonatan Geifman and Ran El-Yaniv. 2017. Selective classification for deep neural networks. *Advances in neural information processing systems*, 30.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR.

Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, and Sai Qian Zhang. 2024. Parameter-efficient fine-tuning for large models: A comprehensive survey. *arXiv preprint arXiv:2403.14608*.

John Hewitt and Percy Liang. 2019. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (emnlp-ijcnlp)*, pages 2733–2743.

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.

Dapeng Hu, Mi Luo, Jian Liang, and Chuan-Sheng Foo. 2024. Towards reliable model selection for unsupervised domain adaptation: An empirical study and a certified baseline. *Advances in Neural Information Processing Systems*, 37:135883–135903.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, and 1 others. 2022a. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Shengding Hu, Ning Ding, Huadong Wang, Zhiyuan Liu, Jingang Wang, Juanzi Li, Wei Wu, and Maosong Sun. 2022b. Knowledgeable prompt-tuning: Incorporating knowledge into prompt verbalizer for text classification. In *Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 2225–2240.

Abhinav Jain, Swarat Chaudhuri, Thomas Reps, and Chris Jermaine. 2024. Prompt tuning strikes back: Customizing foundation models with low-rank prompt adaptation. *Advances in Neural Information Processing Systems*, 37:47297–47316.

Wengong Jin, Regina Barzilay, and Tommi Jaakkola. 2018. Junction tree variational autoencoder for molecular graph generation. In *International conference on machine learning*, pages 2323–2332. PMLR.

Tushar Khot, Ashish Sabharwal, and Peter Clark. 2018. Scitail: A textual entailment dataset from science question answering. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Johannes Kiesel, Maria Mestre, Rishabh Shukla, Emmanuel Vincent, Payam Adineh, David Corney, Benno Stein, and Martin Potthast. 2019. Semeval-2019 task 4: Hyperpartisan news detection. In *Proceedings of the 13th international workshop on semantic evaluation*, pages 829–839.

Wouter M Kouw and Marco Loog. 2019. A review of domain adaptation without target labels. *IEEE transactions on pattern analysis and machine intelligence*, 43(3):766–785.

Marco Laumanns and Jiri Ocenasek. 2002. Bayesian optimization algorithms for multi-objective optimization. In *International Conference on Parallel Problem Solving from Nature*, pages 298–307. Springer.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pages 3045–3059.

Chengzhengxu Li, Xiaoming Liu, Zhaohan Zhang, Yichen Wang, Chen Liu, Yu Lan, and Chao Shen. 2024. Concentrate attention: Towards domain-generalizable prompt optimization for language models. *Advances in Neural Information Processing Systems*, 37:3391–3420.

Linyang Li and Xipeng Qiu. 2021. Token-aware virtual adversarial training in natural language understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 8410–8418.

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.

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin A Raffel. 2022a. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *Advances in Neural Information Processing Systems*, 35:1950–1965.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2022b. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 61–68.

Zeyu Liu, Souvik Kundu, Anni Li, Junrui Wan, Lianghao Jiang, and Peter Beerel. 2024. Aflora: Adaptive freezing of low rank adaptation in parameter efficient fine-tuning of large models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 161–167.

Fang Ma, Chen Zhang, Lei Ren, Jingang Wang, Qifan Wang, Wei Wu, Xiaojun Quan, and Dawei Song. 2022. Xprompt: Exploring the extreme of prompt tuning. *arXiv preprint arXiv:2210.04457*.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2017. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*.

Ruiyao Miao, Junren Xiao, Shiya Tsang, Hui Xiong, and Ying Nian Wu. 2026. Optimizing the unknown: Black box bayesian optimization with energy-based model and reinforcement learning. *Advances in Neural Information Processing Systems*, 38:144585–144637.

Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. 2018. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993.

Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. 2019. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *Advances in neural information processing systems*, 32.

Hoang Phan, Lam Tran, Quyen Tran, and Trung Le. 2024. Enhancing domain adaptation through prompt gradient alignment. *Advances in Neural Information Processing Systems*, 37:45518–45551.

Ahmad Pouramini and Hesham Faili. 2025. Crosspt: Exploring cross-task transferability through multi-task prompt tuning. *arXiv preprint arXiv:2509.14253*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, and 1 others. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, and 1 others. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Abhinav Rastogi, Adam Yang, Albert Q Jiang, Alexander H Liu, Alexandre Sablayrolles, Amélie Héliou, Amélie Martin, Anmol Agarwal, Andy Ehrenberg, Andy Lo, and 1 others. 2025. Devstral: Fine-tuning language models for coding agent applications. *arXiv preprint arXiv:2509.25193*.

Daniela N Rim, DongNyeong Heo, and Heeyoul Choi. 2021. Adversarial training with contrastive learning in nlp. *arXiv preprint arXiv:2109.09075*.

Binxin Ru, Ahsan Alvi, Vu Nguyen, Michael A Osborne, and Stephen Roberts. 2020. Bayesian optimisation over multiple continuous and categorical inputs. In *International Conference on Machine Learning*, pages 8276–8285. PMLR.

Andreas Rücklé, Gregor Geigle, Max Glockner, Tilman Beck, Jonas Pfeiffer, Nils Reimers, and Iryna Gurevych. 2020. Adapterdrop: On the efficiency of adapters in transformers. *arXiv preprint arXiv:2010.11918*.

Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. 2016. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. *Advances in neural information processing systems*, 29.

Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P Adams, and Nando De Freitas. 2015. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175.

Lianghe Shi and Weiwei Liu. 2023. Adversarial self-training improves robustness and generalization for gradual domain adaptation. *Advances in Neural Information Processing Systems*, 36:37321–37333.

Zhengxiang Shi and Aldo Lipani. 2023. Dept: Decomposed prompt tuning for parameter-efficient fine-tuning. *arXiv preprint arXiv:2309.05173*.

Jasper Snoek, Hugo Larochelle, and Ryan P Adams. 2012. Practical bayesian optimization of machine learning algorithms. *Advances in neural information processing systems*, 25.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Shinya Suzuki, Shion Takeno, Tomoyuki Tamura, Kazuki Shitara, and Masayuki Karasuyama. 2020. Multi-objective bayesian optimization using pareto-frontier entropy. In *International conference on machine learning*, pages 9279–9288. PMLR.

Pengwei Tang, Xiaolin Hu, and Yong Liu. 2025. Adept: Adaptive decomposed prompt tuning for parameter-efficient fine-tuning. *arXiv preprint arXiv:2501.03291*.

Antti Tarvainen and Harri Valpola. 2017. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30.

Qwen Team. 2026. Qwen3. 5-omni technical report. *arXiv preprint arXiv:2604.15804*.

Naftali Tishby, Fernando C Pereira, and William Bialek. 2000. The information bottleneck method. *arXiv preprint physics/0004057*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Tu Vu, Brian Lester, Noah Constant, Rami Al-Rfou, and Daniel Cer. 2022. Spot: Better frozen model adaptation through soft prompt transfer. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5039–5059.

A Wang, A Singh, J Michael, F Hill, O Levy, and SR Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. arxiv preprint arxiv: 180407461.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32.

Zhen Wang, Rameswar Panda, Leonid Karlinsky, Rogerio Feris, Huan Sun, and Yoon Kim. 2023. Multitask prompt tuning enables parameter-efficient transfer learning. *arXiv preprint arXiv:2303.02861*.

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122.

Junda Wu, Tong Yu, Rui Wang, Zhao Song, Ruiyi Zhang, Handong Zhao, Chaochao Lu, Shuai Li, and Ricardo Henao. 2023. Infoprompt: Information-theoretic soft prompt tuning for natural language understanding. *Advances in neural information processing systems*, 36:61060–61084.

Jianfei Yang, Hanjie Qian, Yuecong Xu, Kai Wang, and Lihua Xie. 2024. Can we evaluate domain adaptation models without target-domain labels? In *International Conference on Learning Representations*, volume 2024, pages 35061–35081.

Jin Yong Yoo and Yanjun Qi. 2021. Towards improving adversarial training of nlp models. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 945–956.

Kaichao You, Ximei Wang, Mingsheng Long, and Michael Jordan. 2019. Towards accurate model selection in deep unsupervised domain adaptation. In *International Conference on Machine Learning*, pages 7124–7133. PMLR.

Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. 2022. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1–9.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. Paws: Paraphrase adversaries from word scrambling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1298–1308.

Cairong Zhao, Yubin Wang, Xinyang Jiang, Yifei Shen, Kaitao Song, Dongsheng Li, and Duoqian Miao. 2024. Learning domain invariant prompt for vision-language models. *IEEE Transactions on Image Processing*, 33:1348–1360.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *International conference on machine learning*, pages 12697–12706. Pmlr.

Zexuan Zhong, Dan Friedman, and Danqi Chen. 2021. Factual probing is [mask]: Learning vs. learning to recall. *arXiv preprint arXiv:2104.05240*.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. 2022. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16816–16825.

A Implementation, Resources, and Reproducibility

This section reports the implementation and reproducibility details of ARPO from three complementary views. The tables specify the trainable capacity, optimization weights, and default experimental settings used across the main experiments and appendix. Together, they show that ARPO keeps a controlled parameter budget while making the training objective and experimental protocol more transparent and reproducible.

Table 6 reports the trainable modules and parameter allocation of ARPO under $d = 3584$. Most trainable capacity is assigned to reasoning expert residuals and the auxiliary classifier, while the router, conditional generators, token offsets, and DS components remain lightweight. This design keeps the total trainable budget around 18M to 22M parameters, which is small relative to the frozen 9B backbone but directly supports transfer, separation, and confidence control.

Table 7 lists the optimization terms and their default weights. The task loss has weight 1.0, while bottleneck, orthogonality, adversarial invariance, router supervision, balance recovery, and auxiliary support use smaller weights. This setting keeps task learning dominant while enforcing DI and DS separation, source invariance, and stable routing.

Table 8 gives the experimental settings used across backbones and protocols. The prefix length is fixed at $L = 50$, with $L_{\mathrm{DI}} = 25$ and $L_{\mathrm{DS}} = 25$ in the main setting, the learning rate is 5×10^{-5}, and training uses a 20 epoch cap with a three epoch warmup. These shared settings, together with source proxy selection, SWA, and confidence gated decoding thresholds, reduce target specific tuning and support reproducible evaluation.

Artifact Licenses. We use public benchmarks and pretrained backbones following their respective licenses and terms of use. All datasets and pretrained models used in our experiments are used only for research purposes, and no redistribution of the original artifacts is included in this paper.

B Supplementary Experimental

B.1 Full Main Results with Std

Table 9 extends the results in Table 1 with standard deviations, showing that ARPO not only achieves the best accuracy across all CombA settings but also maintains lower variance. Compared with

Table 6: Trainable module shapes and parameter counts. Counts are given for $d = 3584$, the hidden size of Qwen3.5 9B.

Module	Shape	Trainable parameters	Role
Global base B_0	$L_{\mathrm{DI}} \times r_h$, followed by an MLP from r_h to d	about 1.1M	shared DI prefix carrier
Reasoning expert residuals	$R \cdot (L_{\mathrm{DI}} \times r_h)$, with R expert MLPs	about 7.7M when $R = 7$	task family specialization
Reasoning router with learned prior	$d \to 256 \to R$, plus $\log \pi_{\mathrm{prior}} \in \mathbb{R}^R$	about 0.9M	instance dependent expert mixing
Class conditioned residual	basis rank r_c with shape $L_{\mathrm{DI}} \times d$, plus small MLPs	about 0.5M	class aware DI correction
Instance prompt generator	basis rank r_p with shape $L_{\mathrm{DI}} \times d$, plus coefficient MLP	about 0.5M	instance level prefix adaptation
Token adaptive offset	bottleneck rank $r_t = 64$, plus a gate	about 0.5M	token level prefix correction
DS embeddings and projections	$K \cdot (L_{\mathrm{DS}} \times r_h)$, plus projection MLPs	about 1.0M when $K = 12$	source-specific residual information
DS instance conditional generator	basis rank r_d with shape $L_{\mathrm{DS}} \times d$, plus a gate	about 0.6M	instance dependent DS adjustment
Domain and task discriminators	two layer MLPs with spectral normalization	about 0.3M each	adversarial invariance probes
Input gate	$d \to 128 \to L_{\mathrm{DI}}$	about 0.5M	position level DI gating
Verbalizer mixer	$w^{(c)} \in \mathbb{R}^{K_c}$, with $K_c \leq 4$	about 8 scalars	candidate token mixture
Auxiliary classifier	T heads, each with $d \to 256 \to C$	about 5.5M when $T = 6$	detached probability support

Table 7: Loss terms and default weights.

Term	Role	Default weight
$\mathcal{L}_{\mathrm{task}}$	language model task loss on the projected decision vocabulary	1.0
$\mathcal{L}_{\mathrm{vib}}$	information bottleneck penalty on h_{DI}	0.10
$\mathcal{L}_{\mathrm{orth}}$ and HSIC	linear and nonlinear separation between h_{DI} and h_{DS}	0.10, adaptively up to 0.50
$\mathcal{L}_{\mathrm{dom}}$	domain adversarial loss	$0.30 \cdot s_{\mathrm{adv}}$
$\mathcal{L}_{\mathrm{task}}^{\mathrm{adv}}$	task family adversarial loss when enabled	0.30
$\mathcal{L}_{\mathrm{bal}}$	class balance and entropy recovery	0.10
$\mathcal{L}_{\mathrm{type}}^{\mathrm{vib}}$	capacity control for reasoning expert embeddings	0.05
$\mathcal{L}_{\mathrm{rt}}$	router supervision	0.20
$\mathcal{L}_{\mathrm{rH}}$	router entropy hinge and distribution KL	0.05 and 0.05
$\mathcal{L}_{\mathrm{prior}}$	learned prior KL	0.02
$\mathcal{L}_{\mathrm{cons}}$	consistency loss	0.05
$\mathcal{L}_{\mathrm{conc}}$	attention concentration regularizer	0.02
$\lVert \Delta E \rVert_2^2$ and $\lVert P_{\mathrm{DI}}^{\mathrm{gen}} \rVert_2^2$	token offset and generated prefix regularization	0.005 each
$\mathcal{L}_{\mathrm{aux}}$	detached auxiliary classifier loss	0.20

the strongest baselines, ARPO improves the mean accuracy from 81.9, 83.2, and 86.5 to 86.4, 87.3, and 90.7 on 9B, 24B, and 35B, while most standard deviations decrease to 0.3 to 0.7 rather than the 0.7 to 1.1 range seen in baselines. This stability comes from DI and DS prefix separation, which preserves reusable reasoning in the invariant branch while isolating source specific domain and style signals. Source only proxy selection and confidence gated decoding further reduce checkpoint instability and target dependent tuning, explaining why ARPO improves both accuracy and robustness under the no target label protocol.

Table 10 extends the results in Table 2 with standard deviations, showing that ARPO achieves the best accuracy across all CombB settings while keeping lower variance under harder cross task transfer. Compared with the strongest baselines, ARPO improves the mean accuracy from 81.7, 82.6, and 86.1 to 83.1, 84.4, and 88.3 on 9B, 24B, and 35B, while most standard deviations decrease to 0.3 to 0.7 rather than the 0.6 to 1.0 range seen in baselines. This stability comes from DI and DS prefix separation, which preserves reusable reasoning across task families while isolating source specific task and domain signals. Source only proxy selection and confidence gated decoding further reduce checkpoint instability and target dependent tuning, explaining why ARPO improves both accuracy and robustness under the no target label protocol.

B.2 Training Task Analysis

This subsection describes the training and transfer tasks in the two experimental protocols. CombA mainly changes domain, style, or data source while keeping the reasoning type close. CombB holds out the target task family and tests whether the learned prefix can reuse reasoning ability across more distant tasks.

All datasets used in CombA and CombB are public benchmarks rather than newly constructed datasets. The task families are defined as follows: N denotes natural language inference, cov-

Table 8: Default hyperparameters used in ARPO.

Hyperparameter	Value	Note
Backbone	Qwen3.5 9B, Devstral2 24B, Qwen3.5 35B	all under 4-bit quantization
$L_{\mathrm{DI}}, L_{\mathrm{DS}}$	$25, 25$	total length $L = 50$
r_h	256	reparameterizer size
r_p, r_t, r_c, r_d	$4, 64, 4, 8$	instance, token, class, and DS ranks
$s_g, s_t, s_{\mathrm{cls}}, s_d$	$0.08, 0.08, 0.04, 0.02$	residual scales
σ_p	0.01 by default, 0.015 for BoolQ, 0.04 for PAWS robustness runs	prefix noise
p_0	0.15	DS dropout cap
Learning rate	5×10^{-5}	AdamW
Batch size	6 for 9B, 4 for 35B and 24B, with $G_a = 2$	effective size 8 to 12
Epoch cap	20	$T_{\mathrm{warm}} = 3$
γ	10	DANN slope
θ_{ent}	0.30	adversarial cutoff
$\tau_{\perp}, \kappa$	$0.30, 5.0$	orthogonality control
H_*	0.55 by default, 0.70 to 0.85 for many family transfer	router entropy floor
$L_{\mathrm{att}}, \beta_v, n_{\mathrm{conc}}$	$2, 0.2, 4$	last two layers, every four steps
α, T_v, T_a	$0.85, 1, 1$	posterior mixture
$K_{\max}$	4	verbalizer candidate cap
Auxiliary dropout	0.3	out of domain regularization
S_{proxy}	64 per source domain	proxy sample size
ρ_{sat}	0.93	saturation cutoff
K_{SWA}	3	SWA window
$\theta_H, \theta_{\min}$	$0.05, 0.55$	confidence gate thresholds
Selection weights	$(0.6, 0.2, 0.1, 0.1)$ before saturation and $(0.2, 0.4, 0.2, 0.2)$ after saturation	accuracy, entropy, domain and task invariance

Table 9: CombA within-family transfer accuracy (%). CombA trains on datasets from the same task family and tests on a held-out dataset, evaluating domain and style transfer under a fixed reasoning type. Each pair reports 9B, 24B, and 35B backbones (Qwen3.5 9B, Devstral2 24B, Qwen3.5 35B; all under 4-bit quantization). Mean is the arithmetic average across the five transfer pairs. **Bold**: best; underline: second-best.

Method	RTE→SciTail			Yelp→SST 2			News→HP			MNLI→QQP			CoLA→PAWS			Mean		
	9B	24B	35B	9B	24B	35B	9B	24B	35B	9B	24B	35B	9B	24B	35B	9B	24B	35B
LLM only	63.9±1.4	67.8±1.2	70.2±1.1	73.7±1.0	74.5±0.9	79.2±1.1	40.4±1.5	43.7±1.3	45.1±1.2	67.7±1.2	69.1±1.0	72.3±1.1	60.5±1.3	61.2±1.2	66.7±1.1	61.2	63.3	66.7
CrossPT	70.4±1.2	72.1±1.1	75.6±1.0	80.9±1.1	82.3±1.0	85.7±0.9	69.2±1.3	70.3±1.2	74.2±1.1	67.8±1.2	69.2±1.1	71.8±1.0	63.4±1.2	64.8±1.1	67.9±1.0	70.3	71.7	75.0
Soft prompt	70.9±1.1	73.2±1.0	76.3±0.9	87.3±0.9	90.1±0.8	93.2±0.8	70.1±1.2	71.1±1.1	75.5±1.0	75.1±1.0	78.3±0.9	80.2±0.9	68.6±1.1	69.7±1.0	73.1±0.9	74.4	76.5	79.7
ATTEMPT	84.7±1.0	85.1±0.9	88.2±0.8	88.6±0.9	89.3±0.8	94.1±0.8	71.2±1.2	72.4±1.1	75.6±1.0	76.8±1.1	78.1±1.0	80.3±0.9	70.6±1.2	71.2±1.1	75.8±1.0	78.4	79.2	82.8
XPROMPT	75.2±1.1	76.1±1.0	78.2±0.9	90.3±0.8	90.6±0.7	94.3±0.7	75.4±1.1	76.2±1.0	79.1±0.9	84.4±0.9	85.3±0.8	89.3±0.8	73.8±1.0	74.5±0.9	76.8±0.9	79.8	80.5	83.5
InfoPrompt	76.7±1.0	77.1±0.9	80.2±0.8	91.5±0.7	92.4±0.6	95.5±0.6	74.1±1.2	75.2±1.0	79.8±0.9	85.8±0.9	86.8±0.8	90.5±0.8	75.2±1.0	76.6±0.9	79.1±0.8	80.7	81.6	85.0
UAdapter	77.9±0.9	79.2±0.8	82.9±0.8	91.6±0.7	92.8±0.6	96.7±0.5	74.8±1.1	75.5±1.0	79.5±0.9	87.2±0.8	88.1±0.8	91.7±0.7	75.4±1.0	76.8±0.9	79.3±0.8	81.4	82.5	86.0
DEPT	77.4±0.9	79.2±0.8	81.3±0.8	90.1±0.8	92.3±0.7	95.5±0.6	73.5±1.1	74.2±1.0	78.2±0.9	85.2±0.9	87.1±0.8	91.3±0.7	75.1±1.0	75.9±0.9	79.6±0.8	80.3	81.7	85.2
DADEE	78.5±0.8	79.6±0.8	82.8±0.7	91.9±0.7	92.7±0.6	96.8±0.5	75.1±1.0	76.7±0.9	79.6±0.8	87.8±0.8	89.1±0.7	93.3±0.7	76.1±0.9	77.9±0.8	80.1±0.7	81.9	83.2	86.5
CoCoOp	88.1±0.7	88.9±0.7	93.2±0.6	87.4±0.9	88.7±0.8	92.9±0.8	79.4±0.8	80.5±0.7	84.6±0.7	76.1±1.0	77.2±0.9	79.8±0.8	75.1±0.9	76.2±0.8	79.5±0.8	81.2	82.3	86.0
ARPO	**91.3**±0.4	**92.1**±0.4	**95.8**±0.3	**92.7**±0.5	**93.5**±0.4	**96.9**±0.3	**82.6**±0.7	**83.3**±0.6	**86.2**±0.6	**88.3**±0.5	**89.2**±0.5	**93.5**±0.4	**77.2**±1.1	**78.4**±0.9	**81.1**±0.8	**86.4**	**87.3**	**90.7**

ering RTE (Dagan et al., 2005), MNLI (Williams et al., 2018), and SciTail (Khot et al., 2018); S denotes sentiment analysis, covering Yelp and SST-2 (Zhang et al., 2015; Socher et al., 2013); P denotes paraphrase detection, covering QQP (Wang et al., 2018) and PAWS (Zhang et al., 2019); and Q denotes question answering, covering BoolQ (Clark et al., 2019). Topic classification uses AG News (Zhang et al., 2015), acceptability judgment uses CoLA (Warstadt et al., 2019), and hyperpartisan news detection uses HP (Kiesel et al., 2019). In CombB, codes like NSP are family combinations rather than dataset names: NSP means that training uses source datasets from NLI, sentiment analysis, and paraphrase detection, while the target family, such as BoolQ, CoLA, or Topic, is held out during training, tuning, and checkpoint selection.

B.2.1 CombA Training Datasets

CombA evaluates within family transfer. The source and target datasets share a related prediction form but differ in domain, writing style, or data construction. This setting tests whether a method can preserve transferable reasoning without overfitting to the source domain.

RTE→SciTail: Both datasets involve entailment style judgment. RTE contains short premise and hypothesis pairs from mixed textual sources, while SciTail uses science questions and supporting sentences from scientific text. The transfer keeps the entailment decision but shifts from general inference to science oriented reasoning.

Yelp→SST 2: Both datasets are used for sen-

Table 10: CombB cross-task-family transfer accuracy (%). CombB trains across different task families and tests on held-out targets, measuring reusable reasoning across task types. Each pair reports 9B, 24B, and 35B backbones (Qwen3.5 9B, Devstral2 24B, Qwen3.5 35B; all under 4-bit quantization).Mean is the arithmetic average across the six transfer pairs. **Bold**: best; underline: second-best.

Method	NSP→BoolQ			NSP→CoLA			NSP→Topic			SPQ→NLI			NPQ→Sent			NSQ→Para			Mean		
	9B	24B	35B	9B	24B	35B	9B	24B	35B	9B	24B	35B	9B	24B	35B	9B	24B	35B	9B	24B	35B
LLM only	59.9±1.4	60.2±1.3	63.4±1.2	69.5±1.2	70.1±1.1	74.2±1.0	58.4±1.5	60.1±1.3	62.2±1.2	63.9±1.3	64.6±1.2	67.8±1.1	64.3±1.3	65.1±1.2	68.3±1.1	69.5±1.2	70.1±1.1	74.6±1.0	64.3	65.0	68.4
Soft prompt	68.4±1.2	69.1±1.1	72.8±1.0	70.3±1.2	71.5±1.1	74.6±1.0	69.8±1.2	70.8±1.1	73.2±1.0	77.6±1.0	78.3±0.9	81.2±0.9	72.3±1.1	73.7±1.0	77.1±0.9	71.0±1.2	72.8±1.0	76.4±0.9	71.6	72.7	75.9
CrossPT	68.5±1.2	70.0±1.1	73.2±1.0	69.4±1.2	71.2±1.1	73.8±1.0	70.2±1.2	71.9±1.1	74.2±1.0	70.4±1.1	71.5±1.0	73.6±0.9	70.7±1.1	72.0±1.0	74.8±0.9	73.4±1.1	75.1±1.0	77.8±0.9	70.4	72.0	74.6
ATTEMPT	79.2±1.0	80.9±0.9	83.8±0.8	77.2±1.0	79.1±0.9	82.4±0.8	80.2±0.9	81.9±0.8	84.1±0.8	85.1±0.9	86.2±0.8	89.4±0.7	82.5±0.9	83.7±0.8	86.2±0.8	79.2±1.0	80.1±0.9	84.3±0.8	80.6	82.0	85.0
XPROMPT	69.1±1.1	70.2±1.0	73.5±0.9	78.2±0.9	79.3±0.8	82.6±0.8	73.5±1.1	74.1±1.0	75.8±0.9	76.6±1.0	77.5±0.9	79.3±0.8	78.2±1.0	79.4±0.9	81.7±0.8	73.5±1.1	74.1±1.0	78.6±0.9	74.9	75.8	78.6
InfoPrompt	69.6±1.1	70.4±1.0	73.5±0.9	71.0±1.1	72.3±1.0	76.5±0.9	73.9±1.1	74.6±1.0	76.5±0.9	77.0±1.0	78.2±0.9	80.1±0.8	78.8±1.0	79.5±0.9	81.7±0.8	64.1±1.3	65.4±1.2	68.3±1.1	72.4	73.4	76.1
DEPT	70.5±1.0	71.8±0.9	74.3±0.8	72.5±1.0	73.7±0.9	76.2±0.8	73.0±1.0	73.9±0.9	77.1±0.8	77.6±0.9	78.5±0.8	81.8±0.8	79.1±0.9	79.8±0.8	83.7±0.7	68.5±1.1	69.8±1.0	72.2±0.9	73.5	74.6	77.6
UAdapter	72.4±1.0	73.0±0.9	76.1±0.8	74.2±1.0	75.3±0.9	78.8±0.8	74.6±1.0	75.5±0.9	78.9±0.8	81.5±0.9	82.3±0.8	85.4±0.7	76.0±1.0	76.8±0.9	79.9±0.8	65.8±1.2	66.3±1.1	70.0±1.0	74.1	74.9	78.2
DADEE	73.1±0.9	73.9±0.8	75.7±0.8	75.3±0.9	76.1±0.8	80.6±0.7	75.2±0.9	75.8±0.8	79.5±0.7	82.1±0.8	83.6±0.8	86.8±0.7	76.6±0.9	77.2±0.8	80.7±0.7	70.5±1.0	71.3±0.9	75.3±0.8	75.5	76.3	79.8
CoCoOp	84.3±0.7	85.1±0.7	88.5±0.6	76.6±0.9	77.4±0.8	80.3±0.7	78.8±0.8	79.2±0.8	83.1±0.7	88.7±0.7	89.4±0.6	92.6±0.6	83.5±0.8	84.7±0.7	89.2±0.6	78.4±0.9	79.5±0.8	83.0±0.7	81.7	82.6	86.1
ARPO	**85.2**±0.5	**86.7**±0.4	**90.1**±0.3	**79.7**±0.6	**80.6**±0.5	**85.2**±0.4	**80.8**±0.6	**82.1**±0.5	**85.6**±0.4	**89.2**±0.4	**90.5**±0.4	**94.3**±0.3	**84.2**±0.5	**85.9**±0.4	**90.1**±0.3	**79.6**±0.7	**80.4**±0.6	**84.7**±0.5	**83.1**	**84.4**	**88.3**

timent classification. Yelp contains user reviews from daily service and restaurant scenarios, while SST 2 contains shorter movie review sentences. This pair tests whether sentiment knowledge from longer reviews can transfer to a more compact sentiment domain.

News→HP: Both datasets involve news article classification. The News source covers general news topics, while HP focuses on detecting hyperpartisan political writing. This pair creates a clear shift in content style, since the target depends more on partisan framing than ordinary topical cues.

MNLI→QQP: Both datasets require comparing two texts and judging their semantic relation. MNLI emphasizes inference across textual genres, while QQP evaluates whether two Quora questions express the same meaning. This pair keeps semantic comparison but changes the input form from premise hypothesis inference to question pair equivalence.

CoLA→PAWS: Both datasets require sensitivity to linguistic form and sentence meaning. CoLA focuses on grammatical acceptability, while PAWS tests whether sentences preserve meaning after word reordering. This pair examines whether acceptability signals can support adversarial paraphrase judgment.

B.2.2 CombB Training Datasets

CombB evaluates cross task family transfer. Each source code denotes the task families used for training. N, S, P, and Q denote natural language inference, sentiment analysis, paraphrase detection, and question answering. The target family is held out during training, tuning, and checkpoint selection.

NSP→BoolQ: The model is trained on natural language inference, sentiment analysis, and paraphrase detection, then tested on BoolQ. BoolQ requires answering yes or no questions from a passage, so the target depends on evidence grounding rather than sentence comparison or polarity recognition. This setting tests transfer to question answering without target labels.

NSP→CoLA: The model is trained on natural language inference, sentiment analysis, and paraphrase detection, then tested on CoLA. CoLA evaluates grammatical acceptability, where the decision depends more on sentence well formedness than semantic relation or sentiment polarity. This setting tests whether useful linguistic signals remain available beyond the source task families.

NSP→Topic: The model is trained on natural language inference, sentiment analysis, and paraphrase detection, then tested on topic classification. Topic classification requires recognizing the global subject of a text, while the source tasks focus on relation, polarity, or equivalence. This setting measures whether source reasoning can support broader category understanding.

SPQ→NLI: The model is trained on sentiment analysis, paraphrase detection, and question answering, then tested on natural language inference. NLI requires deciding whether a hypothesis follows from, contradicts, or is neutral with respect to a premise. This setting examines whether entailment reasoning can be recovered from affective, equivalence, and passage grounding signals.

NPQ→Sent: The model is trained on natural language inference, paraphrase detection, and question answering, then tested on sentiment analysis. Sentiment analysis requires identifying affective polarity, while the source tasks emphasize relation, equivalence, and evidence grounding. This setting tests transfer to a target where emotional orientation is the main decision signal.

NSQ→Para: The model is trained on natural

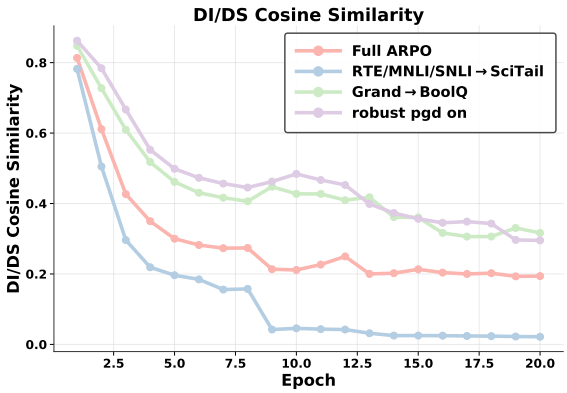

Figure 6: Training cosine similarity over epochs.

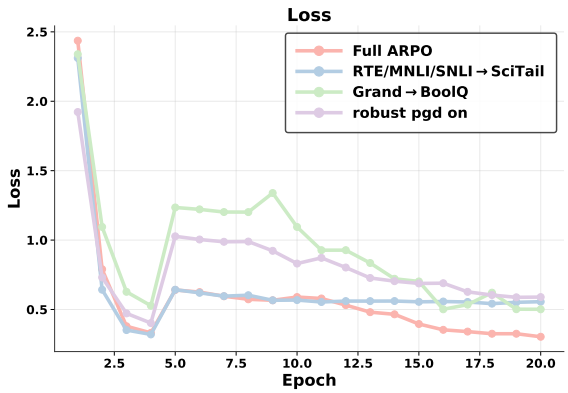

Figure 7: Training loss over epochs.

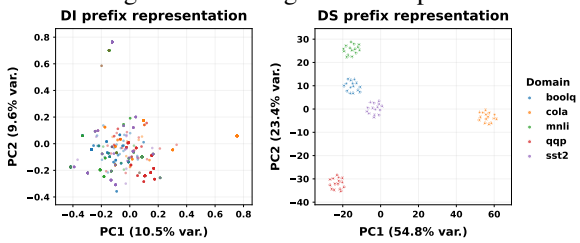

Figure 8: PCA projection of DI and DS prefix representations colored by source domain.

language inference, sentiment analysis, and question answering, then tested on paraphrase detection. Paraphrase detection judges whether two sentences express the same meaning, while the source tasks provide partial signals through inference, polarity, and passage based reasoning. This setting tests reuse of semantic comparison ability under a held out paraphrase family.

B.3 Training Dynamics, Time and Source Proxy Diagnostics

This section provides four diagnostics examining whether ARPO learns stable, transferable prefixes under source only supervision. Figure 6 evaluates DI and DS separation, Figure 7 tracks optimization stability, Figure 10 measures source proxy checkpoint selection reliability, and Figure 11 verifies whether source discriminative ability is preserved. Together, these results show that ARPO can reduce source-specific leakage, maintain stable convergence, improve source proxy signals, and retain source accuracy without target labels. This supports ARPO design, where representation disentanglement, joint optimization, and source proxy selection work together to improve robust transfer.

B.3.1 Representation Separation and Optimization Stability

Figure 6 shows DI and DS cosine similarity declines across all settings, indicating ARPO separates invariant reasoning from source-specific information. SciTail transfer drops fastest and approaches zero, Full ARPO stabilizes around 0.2, and the BoolQ transfer and robust PGD setting remain above 0.3 in later epochs. This pattern follows the orthogonality and information bottleneck constraints, which reduce leakage between DI and DS representations. Higher residual similarity in harder settings shows that cross task transfer and robustness training retain more task specific signals, while ARPO still keeps the two branches separated.

Figure 7 shows that training loss drops quickly in early epochs and reaches stable convergence, indicating efficient prefix adaptation with a frozen backbone. Full ARPO obtains the lowest and smoothest final loss, the SciTail transfer converges early, and the BoolQ transfer and robust PGD setting recover after a middle stage increase. This trend reflects the joint objective, where task learning, adversarial invariance, representation separation, and regularization are optimized together. Later recovery in harder settings shows that adaptive constraints and confidence aware design improve stability under domain shift and perturbation.

Figure 8 visualizes the separation between DI and DS representations. DI representations are mixed across source domains, showing that the invariant prefix suppresses domain specific clustering. In contrast, DS representations form clearer source domain clusters, indicating that the scaffold absorbs source style and domain dependent signals during training. This pattern supports the design goal of storing transferable reasoning in DI while isolating source-specific information in DS.

B.3.2 Source Proxy Selection and Source Retention

Figure 9 shows that Bayesian router entropy gradually decreases during training, indicating that

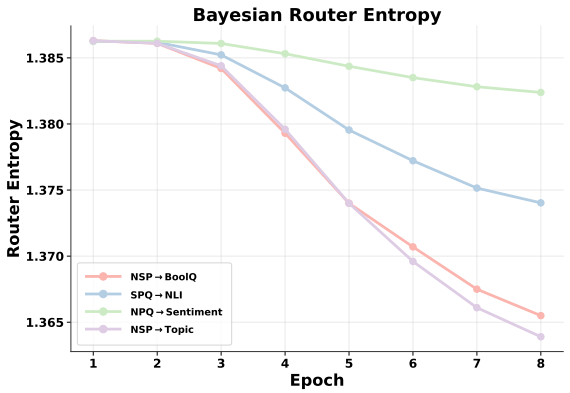

Figure 9: Bayesian router entropy showing confidence growth during training.

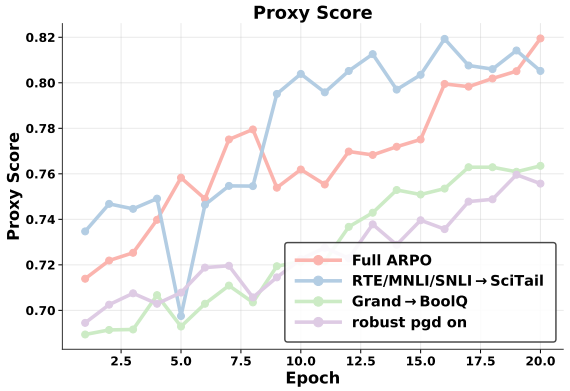

Figure 10: Source proxy score over epochs.

the routing distribution becomes more confident and stable as transferable reasoning patterns are learned. Full ARPO maintains the smoothest entropy decline, while harder cross task settings such as NSP→BoolQ remain at relatively higher entropy, reflecting greater uncertainty when transferring across distant task families. This behavior is consistent with the Bayesian routing design in Sec. 3.1, where the learned prior and adaptive expert allocation progressively separate reusable reasoning from source specific interference. The stable entropy reduction also suggests that the router avoids collapse into overly sharp expert assignment, allowing ARPO to preserve both expert diversity and transfer robustness under the source only protocol.

Figure 10 shows that the source proxy score increases during training and provides a stable source only signal for checkpoint selection. Full ARPO rises smoothly and reaches the highest final score, the SciTail transfer improves quickly in the middle stage, and the BoolQ transfer and robust PGD setting remain lower but continue to improve. Since the proxy score combines source performance, entropy, domain invariance, and task invariance, it reflects selection quality beyond source accuracy alone. The final advantage of Full ARPO supports

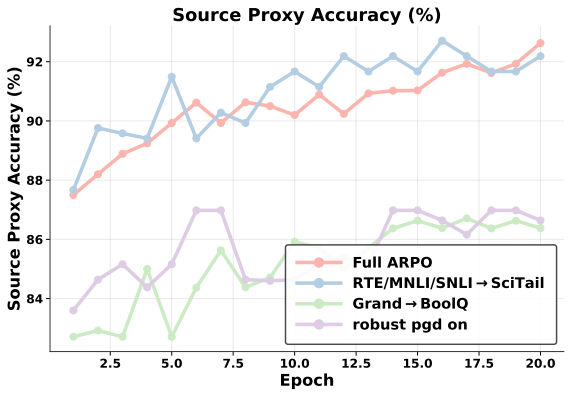

Figure 11: Source proxy accuracy over epochs.

the source only selection rule, which avoids target labels while still identifying checkpoints with stronger transfer potential.

Figure 11 shows stable source proxy accuracy, confirming that ARPO improves transfer without losing source domain discrimination. Full ARPO and the SciTail transfer reach about 92% in later epochs, while the BoolQ transfer and robust PGD setting are lower but steadily improve. This source retention comes from the shared decision interface and auxiliary supervision, which preserve useful decision signals while the DI branch is regularized for transfer. The stable curves in harder settings show that ARPO maintains source competence even when cross task transfer and robustness constraints increase optimization difficulty.

B.3.3 Training Time Analysis

Table 11 and Figure 3 show that ARPO achieves a better efficiency and effectiveness tradeoff than stronger prompt tuning baselines. On the 9B backbone, ARPO uses 22.0 GB memory and 1.6 hours, which is much lower than DePT with 29.5 GB and 3.6 hours and CoCoOp with 31.0 GB and 4.0 hours. This advantage remains on larger backbones: on 24B, ARPO reduces training time from 8.1 and 8.9 hours to 3.7 hours, and on 35B, it reduces training time from 11.4 and 12.5 hours to 5.2 hours. Although ARPO is slightly more expensive than LLM only, the additional cost is small, with only 3.6 GB and 0.4 hours on 9B, while Figure 3 shows that it brings the highest accuracy and robustness with a much lower normalized cost than DePT and CoCoOp. This result supports the design of ARPO, since its router, DI DS separation, and confidence aware gating introduce limited overhead but avoid the heavier optimization cost of prior prompt methods, leading to stronger transfer and robustness under a practical resource budget.

Table 11: Runtime and memory comparison on A6000 with batch size 32. 9B, 24B, and 35B denote Qwen3.5 9B, Devstral2 24B, and Qwen3.5 35B. Memory is reported in GB and time is reported in hours.

Backbone	LLM only		DePT		CoCoOp		ARPO	
	Mem	T	Mem	T	Mem	T	Mem	T
9B	18.4	1.2	29.5	3.6	31.0	4.0	22.0	1.6
24B	35.5	2.8	45.1	8.1	46.3	8.9	39.8	3.7
35B	42.3	4.0	47.2	11.4	47.9	12.5	45.8	5.2

C Theoretical Analysis of ARPO

This appendix provides the theoretical support for the methodology in Sec. 3. The analysis formalizes ARPO as a source-only training-selection-deployment protocol. It does not claim exact disentanglement or target-domain optimality. Instead, it shows how the DI/DS prefix decomposition, progressive scaffold removal, prefix-first gate, source proxy score, and MOBO outer search follow from the same target-label-free transfer constraint.

C.1 Proof Roadmap and Claim Scope

This appendix provides the theoretical support for the methodology in Sec. 3. The goal is to formalize why the three mechanisms in ARPO form a coherent source-only training-selection-deployment loop. We show how the DI/DS prefix split changes the training incentive, how the regularizers act as tractable surrogates for the intended invariance behavior, how progressive DS dropout connects source training to DI-only deployment, how the prefix-first gate can be understood through a risk decomposition, and why the source proxy score and MOBO loop do not introduce target-label leakage.

The proof structure follows the same order as the method. Appendix C.2 defines the source-only transfer protocol and the notion of admissible source-side quantities. Appendix C.3 formalizes the DI/DS split as a training-deployment decomposition: DS provides a source-side scaffold during training, while DI is the only branch deployed at target evaluation. Appendix C.4 explains the information-theoretic target behind the training objective and relates VIB, adversarial probes, and DI-DS separation to probe-based invariance. Appendix C.5 derives progressive DS dropout as an interpolation between source-scaffold training and DI-only deployment. Appendix C.6 studies the prefix-first gate as a source-fixed reliability rule. Appendix C.7 and Appendix C.8 show that checkpoint selection and MOBO search depend only on source-side quantities. Appendix C.9 then states the limits of these claims.

The analysis uses the following protocol conditions and claim boundaries throughout Appendix C.

Condition C.1: Source-only protocol. Target labels are not used for training, checkpoint selection, gate-threshold selection, or MOBO search. All quantities used for model selection are computed from the source training split, the held-out source proxy split, model predictions, or source-side metadata.

Condition C.2: Frozen backbone. Decoder LM f_θ still fixed. Adaptation occurs through the prefix interface, the router, the output interface, and the small modules specified in Sec. 3.1 and Sec. 3.2.

Condition C.3: Source metadata. Source domain labels D and task-type labels T are available for source-side training and probing. No target-domain label is assumed at inference.

Boundary C.4: Probe-based invariance. The analysis does not claim exact statistical independence between h_{DI} and source attributes. Invariance is interpreted through finite-capacity probes and the training surrogates used in Sec. 3.2.

Boundary C.5: Projected decision interface. The shared answer set $\mathcal{A} = \{\mathrm{yes}, \mathrm{no}\}$ defines the decision space used in this paper. For non-binary tasks, the claims apply to the projected interface specified in Appendix A, rather than to the native label space before projection.

Table 12 summarizes the proof route. Each step addresses one potential concern about the methodology: whether the split is more than a naming convention, whether the loss terms are more than engineering additions, whether DS dropout connects training to deployment, whether the gate is more than a threshold heuristic, whether source-only selection leaks target labels, whether MOBO is more than ordinary hyperparameter tuning, and what the theory does not claim.

Together, these results show that ARPO is not a collection of independent stabilizers. The components implement one source-only loop: learn a deployed DI branch under a training scaffold, decide whether to trust it without target labels, and select checkpoints and training recipes using source-side

Step	Role in the proof
Appendix C.2	Defines source-admissible quantities and excludes target labels from training, gating, selection, and MOBO.
Appendix C.3	Formalizes DS as a source-side scaffold and DI as the deployed branch.
Appendix C.4	Links the training regularizers to variational information control and adversarial probe-based invariance.
Appendix C.5	Shows that DS dropout interpolates between source-scaffold training and DI-only deployment.
Appendix C.6	Interprets the prefix-first gate through risk decomposition and source-fixed reliability control.
Appendix C.7	Shows that the checkpoint score is source-admissible.
Appendix C.8	Treats MOBO as outer source-only selection over accuracy, invariance, stability, and cost.
Appendix C.9	States the boundaries: no exact disentanglement, no target-risk guarantee, no oracle gate, and no target-tuned MOBO.

Table 12: Roadmap of the theoretical analysis in Appendix C. The appendix supports the methodology in Sec. 3 without claiming exact disentanglement or target-domain optimality.

quantities only. Appendix C.9 states the limits of these claims.

C.2 Problem Setup and Source-Only Transfer Protocol

This section formalizes the source-only protocol used by ARPO. The purpose is to specify which quantities are admissible under the target-label-free setting and to show that the training, gate-threshold selection, checkpoint selection, and MOBO search rules used in ARPO do not query target labels. This distinction is important in cross-domain evaluation, where target labels are unavailable during selection and model choice can otherwise become an implicit source of leakage (Kouw and Loog, 2019; Yang et al., 2024; Hu et al., 2024).

Source and target data. Let the labeled source collection be

$$\mathcal{D}_{\text{src}} = \{(x_i, y_i, d_i, t_i)\}_{i=1}^{n}, \tag{13}$$

where x_i is an input, y_i is the source label, $d_i \in \mathcal{D}$ is the source domain index, and $t_i \in \mathcal{T}$ is the source task type. The source collection is split into a training subset and a held-out source proxy subset,

$$\mathcal{D}_{\text{src}} = \mathcal{D}_{\text{train}} \cup \mathcal{D}_{\text{proxy}}, \qquad \mathcal{D}_{\text{train}} \cap \mathcal{D}_{\text{proxy}} = \emptyset. \tag{14}$$

The target collection used during deployment is

$$\mathcal{D}_{\text{tgt}} = \{x_j^{\text{tgt}}\}_{j=1}^{m}, \qquad \mathcal{Y}_{\text{tgt}} = \{y_j^{\text{tgt}}\}_{j=1}^{m}, \tag{15}$$

where $\mathcal{Y}_{\text{tgt}}$ is reserved for final reporting only. Under the protocol used in this paper, $\mathcal{Y}_{\text{tgt}}$ is not available for training, threshold selection, checkpoint selection, or outer search.

Let f_θ denote the frozen decoder LM. All trainable quantities are collected into

$$\phi = (\phi_{\text{DI}}, \phi_{\text{DS}}, \rho, \eta),$$

where ϕ_{DI} and ϕ_{DS} parameterize the DI and DS prefix branches, ρ denotes router parameters, and η denotes the output interface and auxiliary modules. The exact module-level parameterization is given in Sec. 3.1 and Sec. 3.2.

Observable source information. For a fixed training run, define the source information set

$$\mathcal{I}_{\text{src}} = \Big(\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{proxy}}, f_\theta, \phi, \{d_i, t_i\}_{i=1}^{n}, \widehat{p}_\phi(\cdot \mid x)\Big), \tag{16}$$

where $\widehat{p}_\phi(\cdot \mid x)$ denotes model predictions or derived confidence statistics on source-side inputs. This set contains the objects used by the task loss, the source-domain and task-type probes, the source proxy score, the gate-threshold search, and the MOBO objectives. It does not contain target labels $\mathcal{Y}_{\text{tgt}}$.

Definition C.1 (Source-admissible quantity)**.** A scalar, vector, checkpoint choice, threshold choice, or candidate-selection rule Q is called *source-admissible* if there exists a measurable map g such that

$$Q = g(\mathcal{I}_{\text{src}}). \tag{17}$$

Equivalently, Q may depend on source training data, the source proxy split, source labels, source domain and task-type metadata, frozen-model outputs, and predictions made by the current prefix interface, but not on $\mathcal{Y}_{\text{tgt}}$.

This definition covers both numerical quantities and discrete choices. For example, a gate threshold selected on $\mathcal{D}_{\text{proxy}}$, a checkpoint selected by Eq. 11,

or a MOBO candidate promoted by the source-side objectives in Eq. 12 is source-admissible if its computation uses only $\mathcal{I}_{\text{src}}$.

Proposition C.2 (Target-label-free admissibility)**.** *Let Q be any quantity used for training, gate-threshold selection, checkpoint selection, or MOBO search. If Q is source-admissible in the sense of Definition C.1, then the value of Q is invariant to any change in the unobserved target labels $\mathcal{Y}_{\text{tgt}}$. Consequently, using Q to choose thresholds, checkpoints, or MOBO candidates cannot introduce target-label leakage.*

Proof. By Definition C.1, there exists a measurable map g such that $Q = g(\mathcal{I}_{\text{src}})$. The information set $\mathcal{I}_{\text{src}}$ contains source-side data, source metadata, frozen-model outputs, and model predictions used during source-side evaluation, but it does not contain $\mathcal{Y}_{\text{tgt}}$. Consider two possible target-label assignments $\mathcal{Y}_{\text{tgt}}$ and $\mathcal{Y}'_{\text{tgt}}$ for the same target inputs, while keeping $\mathcal{I}_{\text{src}}$ fixed. Then

$$Q(\mathcal{I}_{\text{src}}, \mathcal{Y}_{\text{tgt}}) = g(\mathcal{I}_{\text{src}}) = Q(\mathcal{I}_{\text{src}}, \mathcal{Y}'_{\text{tgt}}). \quad (18)$$

Thus Q cannot vary as a function of the unseen target labels. Any procedure that selects a threshold, checkpoint, or candidate configuration using only such Q is therefore target-label-free by construction. □

Consequence for ARPO. The task loss in Eq. 7 is computed from source labels and the projected source answer space. The gate thresholds are selected on the held-out source proxy split. The checkpoint score in Eq. 11 combines gated source-proxy accuracy, source-side predictive uncertainty, and source-side domain/task probes. The MOBO objectives in Eq. 12 are computed from the same source-side accuracy, invariance, stability, and cost quantities. Each of these quantities is source-admissible under Definition C.1; Proposition C.2 therefore applies to the full training-selection-deployment protocol used by ARPO.

The proposition states only the property required by the experimental protocol: training, gating, checkpoint selection, and MOBO search do not depend on target labels. The empirical usefulness of the source proxy score is evaluated separately through the transfer results, ablations, robustness analysis, and source-score diagnostics reported in the main paper and the remaining appendix sections.

C.3 DI/DS Decomposition as Training-Deployment Separation

This section formalizes the DI/DS prefix split used in Sec. 3.1. The purpose is not to prove exact disentanglement, but to show that the split has an operational role under the source-only protocol. Following the motivation of source-invariant representation learning (Ganin et al., 2016; Arjovsky et al., 2019), P_{DI} is the deployed branch, while P_{DS} is a source-side scaffold used only during training. Unlike standard invariant/specific splits, the source-specific branch in ARPO is removed at deployment rather than used as a target adapter. This changes the regularized source objective by giving source-specific variation a training-time outlet outside the deployed branch.

Definition C.3 (Deployed and scaffold prefix branches)**.** A prefix branch is called *deployed* if it remains active at target evaluation. A prefix branch is called a *scaffold* if it can be used during source training but is removed at target evaluation. In ARPO, P_{DI} is deployed, while $P_{\text{DS}}^{(d)}$ is a source-indexed scaffold:

$$P_{\text{train}}(x, d) = \left[P_{\text{DI}}(x); P_{\text{DS}}^{(d)}(x)\right], \quad (19)$$

$$P_{\text{test}}(x) = \left[P_{\text{DI}}(x); 0\right]. \quad (20)$$

To isolate the role of the split, consider the following abstract regularized source objective:

$$\mathcal{J}_{\text{split}} = \mathcal{R}_{\text{src}}(P_{\text{DI}}, P_{\text{DS}}) + \lambda_{\text{inv}}\Omega_{\text{inv}}(P_{\text{DI}}) + \lambda_{\text{sep}}\Omega_{\text{sep}}(P_{\text{DI}}, P_{\text{DS}}),$$

where $\mathcal{R}_{\text{src}}$ is the source task risk under the shared answer interface, Ω_{inv} penalizes recoverable source traces in the deployed branch, and Ω_{sep} penalizes overlap between the DI and DS branches. The concrete training losses in Sec. 3.2 instantiate these terms through VIB, adversarial domain/task probes, DI-DS separation, and stability regularization. Appendix C.4 gives the corresponding surrogate interpretation.

Lemma C.4 (Scaffold reallocation)**.** *Consider two prefix parameterizations, ϕ^{DI} and ϕ^{DS}, that produce the same source-side predictions on $\mathcal{D}_{\text{train}}$. In ϕ^{DI}, a source-specific component $a_d(x)$ is encoded in the deployed branch P_{DI}. In ϕ^{DS}, the same component is encoded in the scaffold branch $P_{\text{DS}}^{(d)}$, while the task-relevant DI component is unchanged. If*

$$\mathcal{R}_{\text{src}}(\phi^{\text{DS}}) \leq \mathcal{R}_{\text{src}}(\phi^{\text{DI}}) \quad (21)$$

and at least one of the regularizers decreases,

$$\Omega_{\text{inv}}(\phi^{\text{DS}}) \leq \Omega_{\text{inv}}(\phi^{\text{DI}}), \tag{22}$$

$$\Omega_{\text{sep}}(\phi^{\text{DS}}) \leq \Omega_{\text{sep}}(\phi^{\text{DI}}), \tag{23}$$

with one strict inequality whenever $\lambda_{\text{inv}} > 0$ *or* $\lambda_{\text{sep}} > 0$*, then*

$$\mathcal{J}_{\text{split}}(\phi^{\text{DS}}) < \mathcal{J}_{\text{split}}(\phi^{\text{DI}}). \tag{24}$$

Proof. Subtracting the two objective values in Eq. C.3 gives

$$\begin{aligned} &\mathcal{J}_{\text{split}}(\phi^{\text{DS}}) - \mathcal{J}_{\text{split}}(\phi^{\text{DI}}) \\ &= \Delta\mathcal{R}_{\text{src}} + \lambda_{\text{inv}}\Delta\Omega_{\text{inv}} + \lambda_{\text{sep}}\Delta\Omega_{\text{sep}}, \end{aligned} \tag{25}$$

where each Δ term denotes the value under ϕ^{DS} minus the value under ϕ^{DI}. By Eq. 21, $\Delta\mathcal{R}_{\text{src}} \leq 0$. By Eq. 23, the weighted regularization difference is non-positive and is strictly negative when a positively weighted regularizer decreases. Therefore the total difference in Eq. 25 is strictly negative, which proves Eq. 24. □

Proposition C.5 (Training-deployment separation)**.** *Under the objective in Eq. C.3, a training-only DS branch reduces the objective-level incentive for the deployed DI branch to encode source-specific components, whenever those components can be represented by DS without increasing source task risk and are penalized by the DI invariance or DI-DS separation terms.*

Proof. The statement follows from Lemma C.4. If a source-specific component can be moved from P_{DI} to $P_{\text{DS}}^{(d)}$ without increasing source risk, and if keeping that component in DI increases either the recoverable source trace penalty or the DI-DS overlap penalty, then the reallocated parameterization has lower regularized objective value. Thus, the split objective favors assigning such source-specific variation to the scaffold branch rather than to the deployed branch. This is an objective-level bias, not a guarantee of exact statistical independence. □

Consequence for ARPO. Proposition C.5 explains why the DS branch is useful during source training even though it is removed at inference. DS gives source-indexed artifacts a branch that is visible to the source objective but absent from target deployment, while the deployed DI branch remains under invariance and separation pressure. The decomposition therefore yields testable predictions rather than a naming convention: DI should remain sufficient at inference, DS should not transfer reliably as a target adapter, and the DI/DS capacity ratio should have an interior optimum. These predictions match the methodology in Sec. 3.1 and are examined through the component and length-ratio analyses in Sec. 4.3 and Table 5, 4. The next section specifies how Ω_{inv} and Ω_{sep} are instantiated through variational information control, adversarial probes, and DI-DS separation.

C.4 Surrogate Information Objective

This section explains why the regularizers in Sec. 3.2 are treated as surrogates for the DI/DS hypothesis rather than as independent engineering additions. The intended behavior is information-theoretic: the deployed representation should remain predictive of the projected label Y, while source domain D and task type T should not be easily recoverable from h_{DI}. This follows the information bottleneck view of preserving task-relevant information under compression (Tishby et al., 2000; Alemi et al., 2016), together with adversarial representation learning for reducing domain recoverability (Ganin et al., 2016). Since exact mutual information is not directly optimized in the neural prefix model, ARPO uses tractable surrogates: task cross-entropy for label sufficiency, VIB for capacity control, gradient-reversal probes for source-trace removal, and DI-DS separation for reducing duplicate encoding across the deployed and scaffold branches.

Let h_{DI} and h_{DS} denote hidden representations induced by the DI and DS branches. The idealized information target can be written as $\underset{\phi}{\text{minimize}} \quad \mathcal{R}_Y(\phi)$ subject to

$$I(h_{\text{DI}}; D) \leq \epsilon_D, \tag{26}$$

$$I(h_{\text{DI}}; T) \leq \epsilon_T, \tag{27}$$

$$I(h_{\text{DI}}; X) \leq \epsilon_X, \tag{28}$$

where $\mathcal{R}_Y$ denotes the prediction risk on the shared answer interface. The constraints state the desired role of DI: it should retain enough information for Y, but should avoid unbounded input memorization and source-attribute recovery. The optimization in Eq. 28 is not solved exactly. Instead, the training objective in Eq. 7 uses the following surrogate terms.

Lemma C.6 (Variational capacity bound)**.** *Let* $q_\phi(h_{\text{DI}} \mid x)$ *be a stochastic encoder for the DI*

representation and let $r(h)$ be a fixed prior. Then

$$I(X; h_{\mathrm{DI}}) \leq \mathbb{E}_x \mathrm{KL}\big(q_\phi(h_{\mathrm{DI}} \mid x) \,\|\, r(h)\big). \quad (29)$$

Thus the VIB penalty upper-bounds the input information carried by h_{DI} up to a non-negative prior-mismatch term.

Proof. Let $q_\phi(h_{\mathrm{DI}}) = \int q_\phi(h_{\mathrm{DI}} \mid x) p(x)\, dx$ be the marginal representation distribution. By the definition of mutual information,

$$I(X; h_{\mathrm{DI}}) = \mathbb{E}_x \mathrm{KL}\big(q_\phi(h_{\mathrm{DI}} \mid x) \,\|\, q_\phi(h_{\mathrm{DI}})\big). \quad (30)$$

For any fixed prior $r(h)$,

$$\begin{aligned} &\mathbb{E}_x \mathrm{KL}\big(q_\phi(h_{\mathrm{DI}} \mid x) \,\|\, r(h)\big) \\ &= I(X; h_{\mathrm{DI}}) + \mathrm{KL}\big(q_\phi(h_{\mathrm{DI}}) \,\|\, r(h)\big). \end{aligned} \quad (31)$$

Since KL divergence is non-negative, Eq. 29 follows. □

Lemma C.6 does not state that VIB alone removes domain information. It states that VIB restricts the total information capacity of the deployed representation. In ARPO, this capacity control is combined with adversarial probes that directly target source domain and task traces.

Definition C.7 (Probe recoverability)**.** For $z \in \{D, T\}$, let $\mathcal{A}_z$ be a finite-capacity probe family and define the best recoverability of z from h_{DI} as

$$a_z^\star = \sup_{a \in \mathcal{A}_z} \Pr\big(a(h_{\mathrm{DI}}) = z\big). \quad (32)$$

Let $b_z = 1/N_z$ be the chance level for N_z source classes. A representation is probe-invariant to z within $\mathcal{A}_z$ when $a_z^\star$ is close to b_z.

Lemma C.8 (Probe-based invariance)**.** *If gradient-reversal training drives the optimal probe accuracy $a_z^\star$ toward the chance level b_z for $z \in \{D, T\}$, then h_{DI} becomes invariant to z in the finite-probe sense used by the source proxy score.*

Proof. By Definition C.7, $a_z^\star$ is the highest accuracy achieved by probes in $\mathcal{A}_z$. If $a_z^\star = b_z$, then no probe in $\mathcal{A}_z$ recovers z better than chance. If $a_z^\star$ approaches b_z, then recoverability of z from h_{DI} decreases within the same probe family. The invariance score in Eq. 11 uses this principle through

$$u_z = \mathrm{clip}_{[0,1]}\left(1 - \frac{a_z - b_z}{1 - b_z}\right),$$

which assigns larger values when the measured source probe accuracy a_z approaches chance. Hence adversarial probe training and the source proxy score use the same probe-based notion of invariance. □

The probe-based statement is deliberately weaker than exact statistical independence. Probe accuracy only measures recoverability within the chosen probe family, and does not rule out information that could be extracted by an unrestricted classifier. This restriction follows the standard caution in interpreting diagnostic probes (Hewitt and Liang, 2019).

Lemma C.9 (DI-DS separation)**.** *Let h_{DI} and h_{DS} be nonzero representations. A squared normalized inner-product penalty*

$$\Omega_{\mathrm{sep}} = \left(\frac{\langle h_{\mathrm{DI}}, h_{\mathrm{DS}} \rangle}{\|h_{\mathrm{DI}}\|_2 \|h_{\mathrm{DS}}\|_2}\right)^2 \quad (33)$$

is minimized when h_{DI} and h_{DS} are orthogonal. Therefore, it discourages duplicate linear encoding across the deployed and scaffold branches.

Proof. By the Cauchy-Schwarz inequality,

$$-1 \leq \frac{\langle h_{\mathrm{DI}}, h_{\mathrm{DS}} \rangle}{\|h_{\mathrm{DI}}\|_2 \|h_{\mathrm{DS}}\|_2} \leq 1.$$

The squared normalized inner product is non-negative and equals zero if and only if $\langle h_{\mathrm{DI}}, h_{\mathrm{DS}} \rangle = 0$. Hence Eq. 33 is minimized when the two representations are orthogonal. □

Proposition C.10 (Surrogate information objective)**.** *The training objective in Eq. 7 implements a tractable surrogate for the information target in Eq. 28: it preserves task-predictive information through the supervised loss, restricts deployed-branch capacity through VIB, reduces source-attribute recoverability through adversarial probes, and discourages duplicate encoding between DI and DS through the separation penalty.*

Proof. The supervised term in Eq. 7 penalizes prediction error on the projected source label $\Psi_t(y)$, and therefore preserves information in h_{DI} that is useful for the shared decision interface. Lemma C.6 shows that the VIB term upper-bounds the input information stored in h_{DI}, which gives a tractable relaxation of the capacity constraint in Eq. 28. Lemma C.8 shows that gradient-reversal probes reduce domain and task recoverability within the probe family used by the source proxy score. Lemma C.9 shows that the DI-DS separation term discourages the same linear signal

from being duplicated in both the deployed and scaffold branches. Combining these terms yields a regularized objective that matches the intended behavior: keep DI predictive for Y, limit its source-specific memorization capacity, reduce recoverable D, T traces, and separate it from DS. □

Claim boundary. Proposition C.10 does not prove exact disentanglement or statistical independence. It establishes a surrogate objective whose components correspond to capacity control, finite-probe invariance, and DI-DS separation. This is the claim used in Sec. 3.1: the DI/DS split is a testable operational hypothesis, and the regularizers in Sec. 3.2 provide tractable pressure toward that hypothesis rather than a closed-form independence guarantee.

C.5 Progressive DS Dropout as Consistency Training

This section explains why progressive DS dropout in Sec. 3.2 is not used as ordinary unit-level dropout. Standard dropout randomly removes hidden units or connections to reduce co-adaptation and overfitting (Srivastava et al., 2014). In ARPO, the mask is applied to the source-specific scaffold branch that is absent at deployment. The resulting objective therefore exposes the deployed DI-only path during source training, which makes the training path closer to the inference path.

For a source example (x, d), let $\mathcal{L}([\cdot])$ denote any source-side prediction loss evaluated through a prefix-conditioned forward path. Additive regularizers that do not depend on the DS mask can be appended after the expectation and do not affect the interpolation below. Define the full scaffold loss and the DI-only deployment loss as

$$\begin{aligned} \mathcal{L}_{\text{full}}(x, d) &= \mathcal{L}\Big(\big[P_{\text{DI}}(x); P_{\text{DS}}^{(d)}(x)\big]\Big), \\ \mathcal{L}_{\text{deploy}}(x) &= \mathcal{L}\big(\big[P_{\text{DI}}(x); 0\big]\big). \end{aligned} \tag{34}$$

The masked objective at training progress e is

$$\mathcal{L}_e^{\text{mask}}(x, d) = \mathbb{E}_{m_{\text{DS}}} \mathcal{L}\Big(\big[P_{\text{DI}}(x); m_{\text{DS}} P_{\text{DS}}^{(d)}(x)\big]\Big), \tag{35}$$

where $m_{\text{DS}} \sim \text{Bernoulli}(1 - \tilde{p}(e))$, as in Eq. 6.

Lemma C.11 (Masked scaffold interpolation)**.** *For the Bernoulli DS mask in Eq. 6, the expected masked loss satisfies*

$$\begin{aligned} \mathcal{L}_e^{\text{mask}}(x, d) = (1 - \tilde{p}(e))\, \mathcal{L}_{\text{full}}(x, d) \\ + \tilde{p}(e)\, \mathcal{L}_{\text{deploy}}(x). \end{aligned} \tag{36}$$

Proof. The mask m_{DS} takes value 1 with probability $1 - \tilde{p}(e)$ and value 0 with probability $\tilde{p}(e)$. When $m_{\text{DS}} = 1$, the prefix path is $[P_{\text{DI}}(x); P_{\text{DS}}^{(d)}(x)]$, which gives $\mathcal{L}_{\text{full}}(x, d)$. When $m_{\text{DS}} = 0$, the prefix path is $[P_{\text{DI}}(x); 0]$, which gives $\mathcal{L}_{\text{deploy}}(x)$. Taking the expectation over the two mask states yields Eq. 36. □

Lemma C.11 gives an exact identity. It shows that DS dropout creates a convex interpolation between the source-scaffold path and the deployment path, rather than merely perturbing hidden units. This connects the construction to consistency training ideas, where stochastic perturbations are used to reduce sensitivity across related forward paths (Sajjadi et al., 2016; Tarvainen and Valpola, 2017).

Proposition C.12 (Progressive scaffold removal)**.** *Assume $p_0 > 0$ in the schedule*

$$\tilde{p}(e) = p_0(0.25 + 0.75 p_e), \qquad p_e \in [0, 1].$$

Then the coefficient of $\mathcal{L}_{\text{deploy}}(x)$ in Eq. 36 increases monotonically with training progress p_e. Consequently, progressive DS dropout shifts the expected source training objective toward the DI-only deployment path.

Proof. The coefficient of $\mathcal{L}_{\text{deploy}}(x)$ in Eq. 36 is $\tilde{p}(e)$. Differentiating the schedule with respect to training progress gives

$$\frac{\partial \tilde{p}(e)}{\partial p_e} = 0.75 p_0. \tag{37}$$

Since $p_0 > 0$, the derivative is positive. Therefore, the expected objective in Eq. 36 assigns increasing weight to the DI-only deployment loss as training progresses. □

Consequence for ARPO. Proposition C.12 explains the role of progressive DS dropout in the training–deployment separation argument. Appendix C.3 shows why DS can serve as a source-side scaffold; the present result shows why this scaffold must be weakened during training. The mask directly exposes the deployed DI path on source data, while still allowing DS to absorb source-specific variation early in optimization. This result does not claim monotonic improvement of source or target accuracy. It only formalizes how the expected training objective assigns increasing weight to the path used at target evaluation.

C.6 Prefix-First Gate as Risk Decomposition

This section formalizes the prefix-first gate in Sec. 3.2. The gate is not an ensemble rule. It is a source-fixed acceptance rule: the prefix path remains the default predictor, while the base LM acts as a fallback when the calibrated reliability evidence is insufficient. This view is related to confidence calibration and selective prediction, where confidence scores are used to decide whether a model output should be trusted (Guo et al., 2017; Geifman and El-Yaniv, 2017; Ovadia et al., 2019). In ARPO, however, the gate is constrained by the source-only protocol in Appendix C.2; thresholds are selected on source-side data and then frozen before target evaluation.

Calibrated-reliability-scores. Let $r \in \{\text{pref}, \text{base}\}$ index the prefix path and the base path. The raw logits of each path determine the class prediction,

$$\hat{y}^r(x) = \arg\max_{c \in \mathcal{A}} \ell_c^r(x). \tag{38}$$

Null-input calibration is used only to compute confidence and entropy:

$$\begin{aligned} \ell^g_{\text{pref}}(x) &= \ell_{\text{pref}}(x) - \ell^{\text{null}}_{\text{pref}}, \\ \ell^g_{\text{base}}(x) &= \ell_{\text{base}}(x) - \ell^{\text{null}}_{\text{base}}. \end{aligned} \tag{39}$$

Let

$$\begin{aligned} q^r(x) &= \max_{c \in \mathcal{A}} \operatorname{softmax}(\ell^g_r(x))_c, \\ H^r(x) &= -\sum_{c \in \mathcal{A}} \operatorname{softmax}(\ell^g_r(x))_c \log \operatorname{softmax}(\ell^g_r(x))_c. \end{aligned} \tag{40}$$

Thus calibration affects the reliability scores used by the gate, but not the raw class identity in Eq. 38.

Prefix acceptance event. The prefix-first gate uses the same acceptance event as Eq. 8:

$$\begin{aligned} \mathcal{B}(x) = \mathbf{1}\{\hat{y}^{\text{pref}}(x) = \hat{y}^{\text{base}}(x)\} \\ \vee \Big[\mathbf{1}\{H^{\text{base}}(x) - H^{\text{pref}}(x) \geq \theta_H\} \\ \cdot \mathbf{1}\{q^{\text{pref}}(x) \geq \theta_{\min}\}\Big]. \end{aligned} \tag{41}$$

The gated predictor is

$$\hat{y}_g(x) = \begin{cases} \hat{y}^{\text{pref}}(x), & \mathcal{B}(x) = 1, \\ \hat{y}^{\text{base}}(x), & \mathcal{B}(x) = 0. \end{cases} \tag{42}$$

The event $\mathcal{B}(x) = 1$ therefore means that the transferred prefix is accepted; otherwise, the frozen base path provides the fallback prediction.

Lemma C.13 (Gate risk decomposition)**.** *Let $R_g = \Pr(\hat{y}_g(X) \neq Y)$ be the risk of the gated predictor. Then*

$$\begin{aligned} R_g = \Pr\left(\mathcal{B}(X) = 1,\ \hat{y}^{\text{pref}}(X) \neq Y\right) \\ + \Pr\left(\mathcal{B}(X) = 0,\ \hat{y}^{\text{base}}(X) \neq Y\right). \end{aligned} \tag{43}$$

Proof. The events $\{\mathcal{B}(X) = 1\}$ and $\{\mathcal{B}(X) = 0\}$ form a disjoint partition of the input space. On the first event, Eq. 42 returns the prefix prediction; on the second event, it returns the base prediction. Applying the law of total probability over this partition gives Eq. 43. □

The two terms in Eq. 43 have different roles. The first term is the error incurred by accepted prefix predictions. The second term is the error incurred when the gate falls back to the bare LM. This decomposition makes explicit that the gate controls when prefix risk is allowed to enter the final decision.

Proposition C.14 (Sufficient condition for non-increasing risk)**.** *For each input x, define the conditional path risks*

$$\begin{aligned} r_{\text{pref}}(x) &= \Pr(\hat{y}^{\text{pref}}(x) \neq Y \mid X = x), \\ r_{\text{base}}(x) &= \Pr(\hat{y}^{\text{base}}(x) \neq Y \mid X = x). \end{aligned} \tag{44}$$

If the gate rejects the prefix only on inputs where the base path has no larger conditional risk,

$$\mathcal{B}(x) = 0 \quad \Longrightarrow \quad r_{\text{base}}(x) \leq r_{\text{pref}}(x), \tag{45}$$

then the gated predictor is no worse than the prefix-only predictor:

$$R_g \leq R_{\text{pref}}, \qquad R_{\text{pref}} = \Pr(\hat{y}^{\text{pref}}(X) \neq Y). \tag{46}$$

Proof. Using Eq. 42, the gated risk can be written as

$$\begin{aligned} R_g = \mathbb{E}\Big[\mathcal{B}(X) r_{\text{pref}}(X) \\ + (1 - \mathcal{B}(X)) r_{\text{base}}(X)\Big]. \end{aligned} \tag{47}$$

The prefix-only risk is

$$R_{\text{pref}} = \mathbb{E}[r_{\text{pref}}(X)]. \tag{48}$$

Subtracting Eq. 48 from Eq. 47 gives

$$\begin{aligned} R_g - R_{\text{pref}} = \mathbb{E}\Big[(1 - \mathcal{B}(X)) \\ \cdot \big(r_{\text{base}}(X) - r_{\text{pref}}(X)\big)\Big]. \end{aligned} \tag{49}$$

Whenever $1 - \mathcal{B}(X) = 1$, condition Eq. 45 implies $r_{\text{base}}(X) - r_{\text{pref}}(X) \leq 0$. Whenever $1 - \mathcal{B}(X) = 0$, the integrand is zero. Therefore the expectation in Eq. 49 is non-positive, which proves Eq. 46. □

Source-fixed approximation. Proposition C.14 states an oracle sufficient condition; ARPO does not assume access to the conditional risks in Eq. 44. Instead, Eq. 41 uses null-calibrated entropy and confidence as operational proxies for path reliability. The thresholds θ_H and $\theta_{\min}$ are selected on the held-out source proxy split and then frozen across transfer pairs, backbones, and reported tables. By Proposition C.2, this threshold-selection rule is source-admissible and cannot introduce target-label leakage.

Claim boundary. The gate analysis does not claim that Eq. 41 satisfies the oracle condition in Eq. 45 on every target input. It characterizes the risk behavior that a reliable fallback rule should approximate. In ARPO, the approximation is implemented through source-fixed confidence and entropy, while the empirical usefulness of the gate is evaluated through the ablations and source-score diagnostics.

C.7 Source Proxy Score as Target-Label-Free Selection

This section formalizes the checkpoint selector used in Sec. 3.2. Appendix C.6 characterizes the risk behavior that the prefix-first gate is designed to approximate. The present section addresses a different question: whether the score used to select checkpoints remains admissible under the source-only protocol in Appendix C.2. This distinction matters because model selection without target labels is a known difficulty in domain adaptation and transfer evaluation (Kouw and Loog, 2019; You et al., 2019; Yang et al., 2024; Hu et al., 2024). The result below is an admissibility statement rather than a target-accuracy consistency theorem.

Source-side score components. Let $\mathcal{D}_{\text{proxy}}$ denote the held-out source proxy split. The gated source accuracy is

$$a_{\text{src}}^{\text{gated}} = \frac{1}{|\mathcal{D}_{\text{proxy}}|} \sum_{(x_i,y_i,t_i)\in\mathcal{D}_{\text{proxy}}} \mathbf{1}\{\hat{y}_g(x_i) = \Psi_{t_i}(y_i)\}, \tag{50}$$

where $\hat{y}_g$ is the gated predictor in Eq. 42, and Ψ_{t_i} is the task-specific projection into the shared answer set. The mean predictive entropy is computed on the same source proxy split:

$$\bar{H}_{\text{pred}} = \frac{1}{|\mathcal{D}_{\text{proxy}}|} \sum_{x_i\in\mathcal{D}_{\text{proxy}}} H(\widehat{p}_\phi(\cdot \mid x_i)). \tag{51}$$

For $z \in \{D, T\}$, let a_z be the accuracy of the corresponding source-side probe, and let $b_z = 1/N_z$ be the chance level. The probe-invariance score is

$$u_z = \text{clip}_{[0,1]}\left(1 - \frac{a_z - b_z}{1 - b_z}\right), \qquad z \in \{D, T\}. \tag{52}$$

As discussed in Appendix C.4, u_D and u_T measure finite-probe recoverability of source attributes from h_{DI}, not exact statistical independence.

The checkpoint score is the deterministic weighted combination

$$\begin{aligned}\text{Score} = w_s a_{\text{src}}^{\text{gated}} + w_e \bar{H}_{\text{pred}} \\ + w_D u_D + w_T u_T.\end{aligned} \tag{53}$$

The weights, saturation rule, and SWA acceptance tolerance are fixed by the source-side protocol and implementation settings reported outside this theoretical appendix. The proof below does not depend on a particular numerical choice of these constants.

Lemma C.15 (Closure under deterministic composition)**.** *Let $Q_1, \ldots, Q_k$ be source-admissible quantities in the sense of Definition C.1. If g is a deterministic measurable map, then*

$$Q = g(Q_1, \ldots, Q_k)$$

is also source-admissible.

Proof. By Definition C.1, each Q_j can be written as $Q_j = g_j(\mathcal{I}_{\text{src}})$ for a measurable map g_j. Therefore

$$Q = g(g_1(\mathcal{I}_{\text{src}}), \ldots, g_k(\mathcal{I}_{\text{src}}))$$

is a measurable function of $\mathcal{I}_{\text{src}}$. Thus Q is source-admissible. □

Proposition C.16 (Source-admissibility of the proxy score)**.** *Under the source-only protocol in Appendix C.2, the score in Eq. 53 is source-admissible. Consequently, checkpoint selection based on this score cannot introduce target-label leakage.*

Proof. The term $a_{\text{src}}^{\text{gated}}$ in Eq. 50 depends only on source proxy inputs, source proxy labels, task projections Ψ_t, and the gated predictor whose thresholds are selected by source-side data. It is therefore source-admissible.

The term $\bar{H}_{\text{pred}}$ in Eq. 51 depends only on model predictions on $\mathcal{D}_{\text{proxy}}$. It is therefore source-admissible.

For $z \in \{D, T\}$, the probe score u_z in Eq. 52 depends on source-side probe accuracy a_z, the chance

baseline b_z, and source metadata. Since source domain and task-type labels are available under Condition C.3 in Appendix C.1, u_D and u_T are source-admissible.

The weighted score in Eq. 53 is a deterministic composition of source-admissible terms. By Lemma C.15, the full score is source-admissible. Proposition C.2 then implies that checkpoint selection based on this score cannot depend on target labels. □

Saturation and SWA. The same argument applies to the saturation switch and late-stage SWA acceptance rule. The saturation condition is a deterministic function of source proxy accuracy, and SWA is accepted only when the source proxy metric is preserved. Both rules therefore remain source-admissible under Lemma C.15. They may affect which checkpoint is selected, but they do not introduce any target-facing criterion.

Claim boundary. Proposition C.16 proves the target-label-free status of the source proxy selector. It does not state that the score is an unbiased estimator of target accuracy, nor does it guarantee that the selected checkpoint is target-optimal. The empirical relationship between the source proxy score and transfer behavior is evaluated through the main transfer results, ablations, robustness analysis, and source-score diagnostics. This boundary is consistent with Appendix C.1: the theorem supports the source-only protocol, while empirical evidence supports the usefulness of the chosen proxy.

C.8 MOBO as Outer Source-Only Selection

This section formalizes the MOBO layer in Sec. 3.3. Bayesian optimization is commonly used for sample-efficient search over expensive black-box objectives (Frazier, 2018; Miao et al., 2026), while multi-objective Bayesian optimization uses Pareto or hypervolume criteria to balance competing objectives (Laumanns and Ocenasek, 2002; Emmerich et al., 2011; Daulton et al., 2020). In ARPO, MOBO is not used as a target-tuned hyperparameter search procedure. It acts as the outer layer of the same source-only selection protocol defined in Appendix C.2: candidate training recipes are evaluated only through source-side accuracy, invariance, stability, and cost.

Let $\omega \in \Omega$ denote a candidate training recipe, where Ω is fixed before target evaluation. The search space includes the learning rate, adversarial warmup length, and the main coefficient groups in Eq. 7, as described in Sec. 3.3. For each candidate, the MOBO objective vector is

$$\mathbf{J}(\omega) = \big(J_{\mathrm{acc}}(\omega), J_{\mathrm{inv}}(\omega), J_{\mathrm{stab}}(\omega), J_{\mathrm{cost}}(\omega)\big), \tag{54}$$

with

$$\begin{aligned} J_{\mathrm{acc}}(\omega) &= 1 - a^{\mathrm{gated}}_{\mathrm{src}}(\omega), \\ J_{\mathrm{inv}}(\omega) &= 1 - \frac{u_D(\omega) + u_T(\omega)}{2}, \\ J_{\mathrm{stab}}(\omega) &= \mathrm{Var}_{e \in \mathcal{E}_{\mathrm{late}}}\big(\mathrm{Score}_e(\omega)\big), \\ J_{\mathrm{cost}}(\omega) &= \log\big(1 + \mathrm{time}(\omega)\big). \end{aligned} \tag{55}$$

These are the same source-side objectives used in Eq. 12. The first measures gated source-proxy error, the second penalizes recoverable source traces in DI, the third penalizes late-stage instability of the source proxy score, and the fourth penalizes unnecessary computational cost.

Definition C.17 (Pareto dominance)**.** For two candidates $\omega_1, \omega_2 \in \Omega$, write $\omega_1 \prec \omega_2$ if

$$J_k(\omega_1) \le J_k(\omega_2) \quad \text{for all } k \tag{56}$$

and the inequality is strict for at least one objective. The source-side Pareto front is

$$\mathcal{P}_{\mathrm{src}} = \left\{\omega \in \Omega : \nexists\, \omega' \in \Omega \text{ s.t. } \omega' \prec \omega\right\}. \tag{57}$$

Lemma C.18 (Source-admissibility of Pareto selection)**.** *If each objective $J_k(\omega)$ is source-admissible for every $\omega \in \Omega$, then the Pareto dominance relation in Definition C.17 and the Pareto front $\mathcal{P}_{\mathrm{src}}$ are source-admissible.*

Proof. For any fixed pair (ω_1, ω_2), the event $\omega_1 \prec \omega_2$ is determined by a finite set of comparisons among the values $J_k(\omega_1)$ and $J_k(\omega_2)$. Since each $J_k(\omega)$ is source-admissible, these comparisons are deterministic functions of source-admissible quantities. The Pareto front in Eq. 57 is obtained by applying these comparisons over candidates in Ω. Therefore, both the dominance relation and the resulting Pareto set depend only on source-admissible objective values. □

Proposition C.19 (MOBO preserves source-only selection)**.** *Assume that the objectives in Eq. 55 are source-admissible and that the final candidate is selected from source-side candidates by Eq. 11. Then the MOBO outer loop cannot introduce target-label leakage.*

Proof. The objective J_{acc} is source-admissible because $a_{\mathrm{src}}^{\mathrm{gated}}$ is computed on the source proxy split. The objective J_{inv} is source-admissible because u_D and u_T are source-side probe quantities. The objective J_{stab} is source-admissible because it is a deterministic function of late-stage source proxy scores, which are source-admissible by Proposition C.16. The objective J_{cost} is source-admissible because runtime is observed during source-side training and does not require target labels.

The MOBO surrogate is fitted only to pairs $(\omega, \mathbf{J}(\omega))$, where $\mathbf{J}(\omega)$ is source-admissible. Candidate proposal by Pareto improvement depends only on the surrogate over these source-side objectives. Candidate promotion to the full training budget depends only on the source-side Pareto relation in Definition C.17. By Lemma C.18, these operations are source-admissible. Finally, the selected configuration is chosen by Eq. 11, which is source-admissible by Proposition C.16. Proposition C.2 therefore implies that the MOBO loop cannot depend on target labels. □

Why MOBO fits the selection problem. The role of MOBO is to organize a source-side tradeoff rather than to optimize target performance directly. When gated source accuracy saturates, candidates can have similar $a_{\mathrm{src}}^{\mathrm{gated}}$ while differing in recoverable source traces, late-stage stability, or computational cost. A single source-accuracy objective cannot express this tradeoff. The four objectives in Eq. 55 therefore separate task utility, invariance, stability, and efficiency, and Pareto search keeps candidates that are not dominated under these source-side criteria.

Claim boundary. Proposition C.19 is an admissibility result, not a convergence theorem for target accuracy. It does not claim that the Pareto front contains the target-optimal configuration, nor that MOBO is guaranteed to outperform fixed defaults in every transfer setting. It states that the outer loop cannot act as a hidden target selector when all objectives, surrogate observations, Pareto decisions, and final selections are source-admissible. The empirical value of the MOBO layer is evaluated through the source-only comparisons and transfer results reported in the experiments and appendix.

C.9 Boundary of Claims

This section fixes the scope of Appendix C. The preceding analysis supports the methodological roles of the components in ARPO under the source-only protocol; it does not replace the empirical evaluation in Sec. 4.

Operational disentanglement. Proposition C.5 and Proposition C.10 support the DI/DS split as an operational training–deployment decomposition. DS provides a source-side scaffold, while DI carries the deployed signal under invariance, capacity, and separation pressure. This does not require exact statistical independence such as $h_{\mathrm{DI}} \perp D$ or $I(h_{\mathrm{DI}}; D) = 0$. The relevant claim is probe-based and objective-level: source attributes should become less recoverable from h_{DI} under the probe family and training surrogates used by the method.

Source-only selection. Proposition C.2, Proposition C.16, and Proposition C.19 establish that thresholding, checkpoint selection, and MOBO search use source-admissible quantities. These results rule out target-label leakage, but they do not turn the source proxy score into an oracle estimator of target accuracy. The gate analysis in Proposition C.14 gives the ideal risk behavior that the source-fixed confidence rule approximates, rather than assuming access to target conditional risks.

Projected decision scope. All claims involving Y, task loss, gated accuracy, and source proxy selection are made under the shared answer interface in Sec. 3.1, where $\Psi_t : \mathcal{Y}_t \to \mathcal{A}$ and $\mathcal{A} = \{\mathrm{yes}, \mathrm{no}\}$. Thus, Appendix C supports ARPO as a source-only training–selection–deployment protocol for the projected discriminative setting studied in this paper. Implementation details remain in Appendix A, and empirical validation remains in Sec. 4.