LEARNING TO ROUTE SUPERVISION ALONG THE PATH: ROUTE OPTIMIZATION FOR GRADUAL DOMAIN ADAPTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Gradual domain adaptation progressively updates a source-trained classifier through unlabeled intermediate domains before reaching the target. However, existing methods often prescribe the update rule or weight schedule in advance, while the reliability of different supervision sources can change along the adaptation path. Source predictions can anchor early updates but become less informative than the adapting model as the domain shifts. Therefore, choosing appropriate supervision at each stage without target labels remains challenging. Based on the above analysis, we introduce PrismBridge, which selects bridge rules that combine frozen source predictions, class-prototype probabilities, and current-model predictions. Specifically, we formulate this choice as route-relaxed path optimization, where fixed-route policies are treated as constrained special cases. Candidates starting from the same state are evaluated using bridge losses, source supervision, and view agreement. A tolerance-aware selector then chooses among near-best candidates according to a predetermined preference order, and source-supervised smoothing stabilizes the selected update. Furthermore, we derive cumulative risk and selection-regret bounds along each policy's route-dependent model trajectory. Experiments across geometric, appearance, temporal, tabular, and corruption shifts demonstrate the effectiveness of PrismBridge, achieving severity-5 accuracy improvements of 8.1 and 15.9 percentage points over the strongest compared baseline on CIFAR-10-C and CIFAR-100-C.

1 INTRODUCTION

Unsupervised domain adaptation transfers knowledge from labeled source data to an unlabeled target distribution (Pan & Yang, 2009). However, in many real-world applications, distribution shifts evolve progressively rather than occurring abruptly, such as changes in lighting, acquisition conditions, or corruption levels. To address such scenarios, gradual domain adaptation (GDA) exploits unlabeled intermediate domains to update the classifier before reaching the final target.

Existing unsupervised adaptation methods mainly focus on learning transferable representations. Distribution matching and adversarial learning seek transferable features by aligning source and target representations (Pan et al., 2011; Tzeng et al., 2014; Long et al., 2015; Ganin et al., 2016; Sun & Saenko, 2016; Tzeng et al., 2017; Long et al., 2017; 2018; Courty et al., 2016; Wang et al., 2025a). However, successful transfer requires not only domain alignment but also preserving class discrimination, label structure, and target-support information (Ben-David et al., 2010; Zhao et al., 2019; Chen et al., 2019; Johansson et al., 2019). Accordingly, transport-based approaches (Damodaran et al., 2018; Li et al., 2020; Kirchmeyer et al., 2022) and intermediate-domain synthesis methods (Abnar et al., 2021; Na et al., 2021; Gong et al., 2019; Dai et al., 2021; Zhuang et al., 2024; Sagawa & Hino, 2025; He et al., 2024; Chen et al., 2026) explicitly construct bridges between source and target distributions. Building on these intermediate domains, GDA performs sequential self-training by transferring predictions between stages, fitting pseudo-labels on newly observed data, and adapting the classifier toward the target using only source labels (Kumar et al., 2020).

Despite these advances, existing GDA methods often rely on manually designed strategies or predefined update schedules. For example, STDW adjusts source and target loss weights according to

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

a fixed schedule during adaptation (Wang et al., 2026). However, the reliability of different supervision sources varies along the adaptation path. For a path with progressively stronger corruption, early stages may benefit from stable source information, whereas later stages may require more target-specific guidance. Meanwhile, pseudo-labels can inherit model bias (Chen et al., 2022a), and repeated fitting of incorrect predictions can reinforce errors through self-training (Arazo et al., 2020). Source hypotheses (Liang et al., 2020) and class prototypes from labeled examples (Snell et al., 2017) provide complementary semantic references. However, after adaptation, the current model may correctly recognize shifted samples that the frozen source model misclassifies, making an unchanged source-heavy strategy insufficient. Therefore, different stages require different combinations of supervision sources. Figure 1 illustrates this route mismatch and motivates adaptive supervision selection based on unlabeled evidence (Hu et al., 2024) while considering both immediate performance and future adaptation effects.

Motivated by these observations, we introduce PrismBridge, a route-relaxed framework that formulates supervision selection as an explicit optimization problem along the adaptation path. Unlike existing GDA approaches with fixed update rules or loss schedules, PrismBridge dynamically determines how frozen source predictions, class prototypes, and current-model predictions contribute to each transition. Our contributions are summarized as follows:

- **Adaptive route optimization for gradual domain adaptation.** We formulate GDA as a route-relaxed path optimization problem, where each transition selects a supervision bridge from a compact candidate library instead of following a predefined adaptation rule. By evaluating candidates on their induced model trajectories, PrismBridge captures the long-term impact of sequential route decisions.
- **Prototype-guided bridge construction and selection.** We develop a unified bridge mechanism that integrates frozen source predictions, source-domain prototype probabilities, and current-model predictions through probabilistic mixtures. Confidence and multi-view agreement filter reliable bridges, while tolerance-aware selection and source-supervised smoothing stabilize adaptation.
- **Trajectory-dependent theoretical analysis.** We establish theoretical guarantees for adaptive route selection, including route regret and target-risk accumulation. The analysis decomposes selection errors into score uncertainty, tolerance deviation, and future trajectory effects, and provides a pathwise risk certificate relating domain shifts, bridge quality, pseudo-label fitting, and sample coverage.
- **Comprehensive evaluation across diverse domain shifts.** We evaluate PrismBridge across geometric, appearance, temporal, tabular, and corruption shifts. Extensive experiments demonstrate consistent improvements over existing GDA and test-time adaptation methods, while ablation studies validate the importance of adaptive route selection and stabilization mechanisms.

2 Preliminaries

2.1 Setting and notation

Let $\mathcal{D}_0, \ldots, \mathcal{D}_T$ be distributions on $\mathcal{X} \times \{1, \ldots, C\}$, with $C \geq 2$. We observe a labeled source sample B_0 and unlabeled samples $B_1, \ldots, B_T$ in their supplied order. There are $T+1$ domains and T transitions. Adaptation uses the unlabeled samples and labeled source replay; labels from $\mathcal{D}_{1:T}$ provide the evaluation targets. A classifier $h_\theta : \mathcal{X} \to \Delta^{C-1}$ returns class probabilities, with encoder $f_\theta : \mathcal{X} \to \mathbb{R}^d$. We seek a final classifier with small target risk $R_T(h) = \mathbb{E}_{\mathcal{D}_T} \ell(h(x), y)$, where ℓ is a stated classification loss. The proofs use the bounded probabilistic error $\ell(p, y) = 1 - p_y$; empirical tables use top-1 accuracy.

2.2 State-dependent stage updates

The adaptation state s_t contains the current parameters, optimizer state, prototypes, and any teacher or memory used by the update. A route specifies how available prediction streams supervise the next stage. Writing $\mathcal{G}_t^r$ for the complete stage update under route r, we have

$$s_{t+1}^\pi = \mathcal{G}_t^{r_t}(s_t^\pi; B_0, B_{t+1}), \qquad r_t = \pi_t(s_t^\pi), \qquad s_0^\pi = s_0. \tag{1}$$

Here $\pi = (\pi_0, \ldots, \pi_{T-1})$ is a route policy. Including the observed history in the state captures the information required by subsequent updates. We condition on the samples and algorithmic randomness when comparing policies. Thus every policy is evaluated on its own recursively generated trajectory. A nonnegative transition certificate $c_t(s, r)$ satisfying $R_{t+1}(h_{\mathcal{G}_t^r(s)}) - R_t(h_s) \le c_t(s, r)$ yields a cumulative risk budget by telescoping. Section S3.6 derives one such certificate; the empirical route score is a computable surrogate for selection.

3 METHODOLOGY

3.1 MOTIVATION ANALYSIS

An ordered domain path changes both the input distribution and the reliability of the learner. Source predictions supply stable class identity, prototypes supply a geometric anchor, and current predictions reflect adaptation already achieved. Their relative usefulness changes along the path as the learner adapts. Figure 1 illustrates the resulting route mismatch: a single mixture can be overly conservative at one stage and overly dependent on target predictions at another. The relevant decision is therefore *which bridge should supervise this stage, given the state reached so far?*

3.2 A ROUTE-RELAXED PATH PROGRAM

Let $\mathcal{R} = \{1, \ldots, K\}$ be a finite route library, Π all finite-horizon policies choosing routes in $\mathcal{R}$, and $\Pi_{\text{fixed}} = \{\pi^k : \pi_t^k(s) = k \ \forall (t, s)\}$ the constant-route policies. With states defined by Equation (1), set

$$J(\pi) = R_0(h_0) + \sum_{t=0}^{T-1} c_t(s_t^\pi, \pi_t(s_t^\pi)), \quad J^*_{\text{relax}} = \inf_{\pi \in \Pi} J(\pi), \quad J^*_{\text{fixed}} = \min_{k \in \mathcal{R}} J(\pi^k). \tag{2}$$

The relaxation removes the requirement to reuse one route. It retains all effects of earlier decisions on later models and costs.

Proposition 3.1 (Relaxation with route-dependent trajectories)**.** *If every constant-route policy is admissible, then $J^*_{\text{relax}} \le J^*_{\text{fixed}}$. If, additionally, costs are evaluated on an exogenously fixed state sequence $\bar{s}_t$, the conditional program has the lower-envelope form*

$$\min_{z_t \in \Delta^{K-1}} \sum_{t,k} z_{t,k} c_t(\bar{s}_t, k) = \sum_t \min_k c_t(\bar{s}_t, k) \le \min_k \sum_t c_t(\bar{s}_t, k). \tag{3}$$

Here z_t is a probability vector over routes. Policy-set inclusion compares complete adaptation paths, while linear minimization over a simplex yields the conditional envelope at fixed states. Thus PrismBridge relaxes the fixed-route constraint while preserving each policy's induced trajectory. Appendix S3.1 gives the derivation and equality condition.

The optimal continuation explicitly accounts for future states. With $V_T^*(s) = 0$, define

$$Q_t^*(s, r) = c_t(s, r) + V_{t+1}^*(\mathcal{G}_t^r(s)), \qquad V_t^*(s) = \min_{r \in \mathcal{R}} Q_t^*(s, r). \tag{4}$$

Thus an optimal stage decision minimizes the cost-to-go Q_t^*, which includes the consequence of the update. PrismBridge implements a tractable local selector and controls its deviation through score tolerance, prediction stability, and source supervision. Section 3.5 makes the resulting approximation terms explicit.

3.3 STABLE PROTOTYPE BRIDGING AND ROUTE SELECTION

Prediction streams and routes. The frozen source model supplies $q_s = h_0$. In its feature space, source prototypes are $p_c^s = |B_{0,c}|^{-1} \sum_{(x,y) \in B_{0,c}} f_0(x)$, where $B_{0,c}$ contains source examples of class c. The prototype and current-model streams are

$$q_p(c \mid x) = \frac{\exp(-\|f_0(x) - p_c^s\|_2^2 / \tau_p)}{\sum_{j=1}^C \exp(-\|f_0(x) - p_j^s\|_2^2 / \tau_p)}, \qquad q_t(\cdot \mid x) = h_t(x), \tag{5}$$

where $\tau_p > 0$ is the prototype temperature. Using the same frozen feature coordinates for samples and prototypes keeps distances well defined throughout adaptation. A route r is the tuple

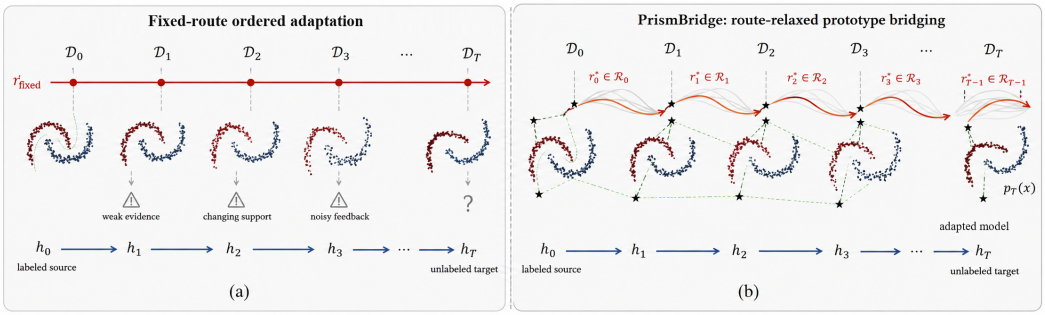


Figure 1: Route decisions along an ordered path. (a) A fixed update rule. (b) Candidate routes (gray), selected routes (colored), prototype anchors (black stars), and class relations (green dashed links).

$(\alpha_r, \beta_r, \gamma_r, \mathcal{V}_r, \kappa_r)$: three nonnegative mixture weights summing to one, a set of label-preserving views, and a consensus rule. Source-anchored, prototype-guided, and target-driven routes place their largest weight on q_s, q_p, and q_t, respectively; a mixed route combines all three. Each route name describes the semantic emphasis of its bridge mixture along the supplied domain path.

For fixed streams, define $w_r = (\alpha_r, \beta_r, \gamma_r)$ and $q_1 = q_s, q_2 = q_p, q_3 = q_t$. The bridge is the unique probability-space barycenter

$$\widetilde{q}_r = \alpha_r q_s + \beta_r q_p + \gamma_r q_t = \arg\min_{q \in \Delta^{C-1}} \frac{1}{2} \sum_{j=1}^{3} w_{r,j} \|q - q_j\|_2^2. \tag{6}$$

Appendix S3.4 derives this closed-form solution by completing the square. Each candidate is therefore constructed directly from the three prediction streams.

Losses with explicit supervision. For a target batch $x_i \in B_{t+1}$, let $\widetilde{y}_{ri} = \arg\max_c \widetilde{q}_r(c \mid x_i)$ and let m_{ri} retain samples passing the confidence and route-consensus tests in Appendix S2.2. Define $M_r = \max(1, \sum_i m_{ri})$. With detached bridge targets, the pseudo-label and alignment losses are

$$L^r_{\text{pseudo}}(\theta) = -\frac{1}{M_r} \sum_i m_{ri} \log h_\theta(x_i)_{\widetilde{y}_{ri}}, \qquad L^r_{\text{align}}(\theta) = \frac{1}{M_r} \sum_i m_{ri} \, \mathrm{KL}(\mathrm{sg}[\widetilde{q}_r(x_i)] \| h_\theta(x_i)). \tag{7}$$

The former fits the class decision; the latter retains the soft bridge distribution. For normalized class centroids $\bar{p}^r_c(\theta)$ weighted by retained bridge probabilities, define $G^r_{cd}(\theta) = \|\bar{p}^r_c(\theta) - \bar{p}^r_d(\theta)\|_2^2$. The topology loss compares these class relations with source relations G^s_{cd}:

$$L^r_{\text{topo}}(\theta) = \frac{1}{\max(1, |\mathcal{P}_r|)} \sum_{(c,d) \in \mathcal{P}_r} \left(G^r_{cd}(\theta) - G^s_{cd}\right)^2, \quad L^r_{\text{bridge}} = \lambda_p L^r_{\text{pseudo}} + \lambda_a L^r_{\text{align}} + \lambda_g L^r_{\text{topo}}. \tag{8}$$

Here $\mathcal{P}_r$ contains class pairs with nonzero retained support; empty sums are zero. Appendix S2.2 specifies the centroids, masks, and normalization. The coefficients $\lambda_p, \lambda_a, \lambda_g$ are nonnegative.

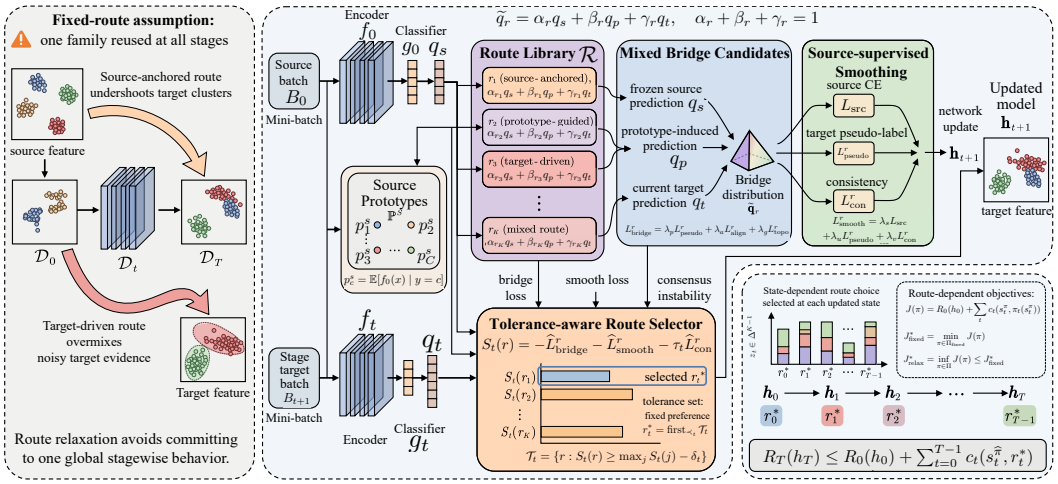
For V views $a_v(x)$, let $p_{iv} = h_\theta(a_v(x_i))$ and $\bar{p}_i = V^{-1} \sum_v p_{iv}$. Multi-view agreement uses generalized Jensen–Shannon divergence:

$$L^r_{\text{con}}(\theta) = \frac{1}{M_r V} \sum_{i,v} m_{ri} \, \mathrm{KL}(p_{iv} \| \bar{p}_i), \qquad L^r_{\text{smooth}} = \lambda_s L_{\text{src}} + \lambda_u L^r_{\text{pseudo}} + \lambda_v L^r_{\text{con}}, \tag{9}$$

where $L_{\text{src}} = -|B_0|^{-1} \sum_{(x,y) \in B_0} \log h_\theta(x)_y$. Source supervision supplies labels independent of the current target predictions; agreement suppresses view-dependent updates. Both are applied with the same loss definitions across routes.

Score tolerance and deterministic route preference. Figure 2 separates candidate scoring from the final route choice. Let $\widehat{L}$ denote losses measured by the common candidate-evaluation procedure. Using the agreement loss in Equation (9), define the larger-is-better score

$$S_t(r) = -\widehat{L}^r_{\text{bridge}} - \widehat{L}^r_{\text{smooth}} - \tau_t \widehat{L}^r_{\text{con}}, \qquad \tau_t \geq 0. \tag{10}$$



(a) Fixed-route mismatch (b) Route-relaxed bridge-and-smooth stage (c) Ordered path optimization

Figure 2: PrismBridge overview. (a) Fixed-route adaptation. (b) Mixed-bridge construction, tolerance-aware route selection, and source-supervised smoothing. (c) Route-dependent model trajectory and risk certificate.

Let $\prec_t$ be a strict total preference order on the candidate identities, fixed by the configuration before adaptation and independent of their measured scores. The selector is

$$\mathcal{T}_t(\delta_t) = \{r : S_t(r) \geq \max_j S_t(j) - \delta_t\}, \qquad r_t^* = \text{first}_{\prec_t} \mathcal{T}_t(\delta_t). \tag{11}$$

Here $\delta_t \geq 0$ determines which score differences exclude a candidate; the preference order resolves the remaining choice. At zero tolerance the selector chooses a score maximizer, including deterministic resolution of exact ties. With illustrative scores $(-0.20, -0.22)$ and preference $2 \prec_t 1$, it selects route 1 at $\delta_t = 0$ and route 2 at $\delta_t = 0.03$. Appendix S3.3 derives the exact switching thresholds and their sensitivity to score perturbations. When a candidate retains zero target samples, $M_r = 1$ and the update consists of source supervision.

3.4 Overall Workflow

Algorithm 1 follows the sequence in Figure 2. All candidates at a stage start from the same s_t, share evaluation batches and views, and use the same candidate budget. The selected candidate initializes the next transition, so each policy generates its own trajectory through controlled stagewise decisions. The deployed model is the final classifier h_T; the route library is used during adaptation.

3.5 Theoretical Analysis

The analysis connects local selection to the cumulative cost of the resulting trajectory. Let $\widehat{c}_t = -S_t$ and assume uniform score calibration: $|\widehat{c}_t(s, r) - c_t(s, r)| \leq \epsilon_t$ for every candidate at the evaluated state.

Theorem 3.1 (Tolerance and trajectory-dependent regret)**.** *The route selected by Equation (11) satisfies*

$$c_t(s_t, r_t^*) - \min_r c_t(s_t, r) \leq \delta_t + 2\epsilon_t. \tag{12}$$

Let $\Omega_t(s) = \max_r V_{t+1}^*(\mathcal{G}_t^r(s)) - \min_r V_{t+1}^*(\mathcal{G}_t^r(s))$. *Then the selected policy* $\widehat{\pi}$ *obeys*

$$J(\widehat{\pi}) - J_{\text{relax}}^* \leq \sum_{t=0}^{T-1} [\delta_t + 2\epsilon_t + \Omega_t(s_t^{\widehat{\pi}})]. \tag{13}$$

The continuation range Ω_t accounts for the effect of a present route on all later costs. It vanishes in the stage-separable case; otherwise the bound retains it. If the score instead approximates Q_t^*, the same proof removes Ω_t. Appendix S3.2 provides the complete telescoping argument.

Algorithm 1 PrismBridge: one route decision per ordered transition

1: Train h_0 on B_0; compute P^s and initialize s_0.
2: **for** $t = 0, \dots, T-1$ **do**
3: Compute shared prediction streams on B_{t+1} and its views.
4: Construct $\widetilde{q}_r$, masks, and candidates from the same s_t for each $r \in \mathcal{R}$.
5: Evaluate $S_t(r)$; form $\mathcal{T}_t(\delta_t)$ and select r_t^* by Equation (11).
6: Commit the selected bridge and source-supervised smoothing update to obtain s_{t+1}.
7: **end for**
8: Return h_T.

Theorem 3.2 (Pathwise risk certificate). *Let* $D_t(h) = |R_{t+1}(h) - R_t(h)|$, $b_t = \mathbb{E}_{P_{t+1}} \mathrm{TV}(\widetilde{q}_{r_t^*}, h_t)$, *and* $\rho_t = \Pr_{P_{t+1}}(m_{r_t^*} = 1)$, *where* P_t *is the input marginal of* $\mathcal{D}_t$. *If* $a_t = \mathbb{E}[\mathrm{KL}(\widetilde{q}_{r_t^*} \| h_{t+1}) \mid m_{r_t^*} = 1]$, *then*

$$R_T(h_T) \le R_0(h_0) + \sum_{t=0}^{T-1} \left[D_t(h_t) + b_t + \rho_t \sqrt{a_t/2} + 1 - \rho_t \right]. \tag{14}$$

The term $\rho_t \sqrt{a_t/2}$ *is zero when* $\rho_t = 0$.

The terms separate domain change, bridge displacement, bridge fitting, and uncovered samples. Evaluating these terms along the selected route-dependent path gives a cumulative bound on the final target risk. Appendix S3.6 provides the one-step derivation and the extension to perturbed domain orders.

Domain order and early decisions. The policy comparison applies to any fixed supplied order because it compares feasible policies under that order. The attainable costs still depend on which domains are supplied and where they occur. Under Lipschitz update operators, Equation (S57) expresses a state perturbation as a sum of earlier route perturbations multiplied by the sensitivities of subsequent updates. Equation (S59) adds the effect of changing an intermediate domain or its position. An early detour can therefore influence several later stages, while a small local perturbation remains controlled when the subsequent updates are stable. This connects the ordered-domain assumption to the early-route interventions in Section 4.3.

4 Main Experimental Results

4.1 Experimental Setup

We evaluate ordered adaptation on Rotated MNIST, Color-Shift MNIST, Portraits, and Covertype (LeCun et al., 1998; Ginosar et al., 2015; Blackard & Dean, 1999). These benchmarks cover geometric, appearance, chronological, and tabular shifts. For CIFAR-10-C and CIFAR-100-C (Hendrycks & Dietterich, 2019), clean data supply the source, severity levels 1–4 supply intermediate domains, and severity 5 is the target. Each corruption path is evaluated separately. Predictive performance is reported as accuracy in percent, with higher values better; accuracy equals 100 minus percentage error. Adaptation and model selection use labeled source data and unlabeled intermediate domains; target labels provide the final evaluation. PrismBridge uses labeled source replay during adaptation.

Image models use CNN or ResNet encoders according to the benchmark configuration; Covertype uses an MLP. Optimization uses Adam, weight decay 10^{-4}, and gradient clipping at 5.0, on NVIDIA RTX 5090 GPUs. The route-library study evaluates $K \in \{1, 2, 3, 4, 6\}$. Appendix S2 specifies losses and candidate evaluation. Appendix S4 details the protocols in Tables S9 and S10, together with five-seed sensitivity means and confidence bands, including the path-density study in Figure S1.

4.2 Baseline Comparison Results

Table 1 compares PrismBridge with standard and ordered domain-adaptation methods, including STDW. PrismBridge attains 96.5%, 99.2%, 93.4%, and 82.0% accuracy, leading the comparison on Color-Shift MNIST, Portraits, and Covertype. On these three tasks, it exceeds STDW, the strongest listed alternative, by 0.9, 6.3, and 7.8 percentage points. Relative to GST, the four gains are 12.7, 25.2,

Table 1: Ordered-benchmark accuracy (%). A dash denotes an unreported result.

Setting	Method	Rotated	Color-shift	Portraits	Covertype
One-shot	DANN (Ganin et al., 2016)	44.2	56.5	73.8	65.2
	Deep CORAL (Sun & Saenko, 2016)	49.6	63.5	71.9	66.8
	DeepJDOT (Damodaran et al., 2018)	51.6	65.8	72.5	67.4
Ordered	GST (Kumar et al., 2020)	83.8	74.0	82.6	73.5
	IDOL (Chen & Chao, 2021)	87.5	78.3	85.5	72.1
	GGF (Zhuang et al., 2024)	67.7	73.6	86.2	71.2
	GOAT (He et al., 2024)	86.4	91.8	83.6	69.9
	AST (Shi & Liu, 2023)	90.6	–	84.8	–
	CNF (Sagawa & Hino, 2025)	62.6	70.4	84.6	70.8
	STDW (Wang et al., 2026)	**97.6**	98.3	87.1	74.2
	PrismBridge	96.5	**99.2**	**93.4**	**82.0**

Table 2: CIFAR-10-C target accuracy (%, ↑) at severity 5. Best values are bold.

Corruption	Source	BN-1	Tent	AdaCon.	CoTTA	GTTA	GST	GOAT	STDW	Ours
Gaussian noise	27.7	71.9	75.2	70.9	75.7	76.6	50.0	27.3	78.5	**89.4**
Shot noise	34.3	73.9	79.4	77.5	78.7	81.7	56.1	34.3	81.0	**90.4**
Impulse noise	27.1	63.7	71.4	70.0	73.4	74.5	49.7	27.0	72.1	**84.3**
Defocus blur	53.1	87.2	85.6	86.0	88.4	89.9	79.4	53.3	90.4	**95.7**
Glass blur	45.7	64.7	68.9	67.3	72.4	72.7	48.8	45.5	71.7	**88.7**
Motion blur	65.2	85.8	83.5	85.9	87.8	88.4	82.8	65.7	89.7	**97.1**
Zoom blur	58.0	87.9	85.9	88.0	89.7	89.9	83.3	58.5	91.5	**99.6**
Snow	74.9	82.7	80.9	83.4	85.2	85.9	82.5	75.1	**97.0**	95.3
Frost	58.7	82.6	81.4	85.1	85.9	87.0	75.7	59.0	86.7	**93.9**
Fog	74.0	84.7	81.4	85.6	87.6	89.1	82.5	74.0	88.3	**94.9**
Brightness	90.7	91.6	87.8	91.9	92.5	92.6	93.1	90.7	92.9	**98.9**
Contrast	53.3	87.4	79.7	90.0	89.4	91.0	86.8	53.4	91.2	**98.4**
Elastic transform	73.4	76.2	74.3	78.1	81.7	80.6	75.1	73.6	80.4	**85.4**
Pixelate	41.5	80.3	79.2	82.3	86.6	85.5	60.1	41.9	86.6	**93.8**
JPEG	69.7	72.7	75.1	80.0	82.7	80.2	73.4	69.8	79.9	**93.9**
Mean	56.5	79.6	79.3	81.5	83.8	84.4	72.0	56.6	85.2	**93.3**

10.8, and 8.5 points. The improvements span appearance, temporal, and tabular shifts, supporting the usefulness of stage-dependent bridges beyond one image transformation.

Tables 2 and 3 report every corruption type at severity 5. PrismBridge reaches mean accuracies of 93.3% on CIFAR-10-C and 90.2% on CIFAR-100-C, compared with 85.2% and 74.2% for STDW. The gains, computed from the unrounded means, are 8.1 and 15.9 percentage points. PrismBridge leads in 14 of the 15 corruption types on CIFAR-10-C and all 15 on CIFAR-100-C, spanning noise, blur, weather, and digital changes. Table S5 summarizes the methods and their access to intermediate severities.

Table 2 reports 89.4% for PrismBridge versus 78.5% for STDW under CIFAR-10-C Gaussian noise; Table 3 reports 97.8% versus 77.9% under CIFAR-100-C defocus blur. The component interventions below examine class anchoring and adaptable bridge mixtures.

4.3 Ablation Studies

Effect of route selection. Table 4 removes one component at a time. Fixing the route while retaining the remaining components lowers accuracy from 96.47 to 93.92 on Rotated MNIST, from 99.20 to 96.02 on Color-Shift MNIST, and from 93.35 to 89.40 on Portraits. These reductions quantify the contribution of stagewise route selection. Figure 3 reinforces this finding: increasing K from one to four improves all four datasets, with little additional benefit from six routes. Thus a small, varied library is sufficient to capture much of the observed route benefit.

Table S8 compares route policies on six-domain Rotated MNIST using the same stabilizers, five candidates, and optimization budget. Adaptive, greedy, and two suitable fixed policies obtain 96.39–

Table 3: CIFAR-100-C target accuracy (%, ↑) at severity 5. Best values are bold.

Corruption	Source	BN-1	Tent	AdaCon.	CoTTA	GTTA	GST	GOAT	STDW	Ours
Gaussian noise	27.0	57.9	62.8	57.7	59.9	63.6	50.2	26.6	69.0	**80.9**
Shot noise	32.0	59.3	64.2	63.2	62.3	67.9	43.3	32.1	72.4	**83.7**
Impulse noise	60.6	57.3	58.3	61.4	60.3	66.0	67.7	60.9	73.9	**85.1**
Defocus blur	70.7	72.4	62.1	72.3	73.1	75.6	77.5	71.3	77.9	**97.8**
Glass blur	45.9	58.1	48.8	59.9	62.0	64.8	58.4	46.2	68.4	**83.1**
Motion blur	69.2	70.3	51.7	70.9	72.1	74.1	75.0	69.8	77.1	**95.7**
Zoom blur	71.2	72.1	51.5	72.5	73.6	76.1	76.7	71.3	77.4	**99.6**
Snow	60.5	65.1	41.6	67.1	67.2	71.1	69.7	60.7	75.0	**90.0**
Frost	54.2	65.0	36.3	69.3	68.2	72.5	67.8	54.3	74.6	**89.7**
Fog	49.7	58.5	28.9	61.8	59.7	69.1	61.9	50.0	74.2	**84.9**
Brightness	70.5	73.5	29.6	74.1	75.3	77.4	77.9	70.6	78.1	**99.3**
Contrast	44.9	69.7	17.7	71.7	73.1	76.6	73.0	46.3	75.7	**99.3**
Elastic transform	62.8	64.3	12.0	66.1	67.5	70.6	66.9	63.2	72.8	**81.8**
Pixelate	25.3	67.1	11.5	66.7	71.7	74.5	59.2	25.7	77.1	**92.8**
JPEG	58.8	58.8	9.6	63.8	66.5	66.7	64.2	58.8	70.0	**89.1**
Mean	53.6	64.6	39.1	66.6	67.5	71.1	66.0	53.9	74.2	**90.2**

Table 4: Component ablations: target accuracy (%).

Variant	Rotated	Color-shift	Portraits
Full	**96.47**	**99.20**	**93.35**
No topology	95.63	98.21	91.82
Fixed route	93.92	96.02	89.40
No source supervision	94.68	96.71	88.70
No view agreement	95.16	97.48	90.95
No mixed bridge	94.93	97.02	91.15
Global tolerance	95.45	97.86	91.55

Table 5: Route timing: accuracy (%).

Route schedule	Portraits	Covertype
Full	93.35	82.02
Fixed early	88.94	76.84
Fixed middle	90.71	78.93
Fixed late	91.82	79.75
Only early selected	90.05	77.96
Only late selected	86.47	75.31

96.48%; random selection and fixed sharpening obtain 9.74% and 9.87%. The dependence on route policy persists with identical stabilizers. Suitable fixed policies on this regular path complement the temporal interventions below, which test changing route suitability.

Role of the stabilizing components. In Table 4, removing source supervision causes the largest Portraits reduction of 4.65 points. True source labels provide a persistent class reference as target predictions evolve. Removing route selection causes the largest reductions on the two MNIST tasks. Topology preservation contributes 0.84–1.53 points by constraining inter-class geometry; view agreement contributes 1.31–2.40 points by discouraging perturbation-dependent predictions; mixed bridging contributes 1.54–2.20 points by retaining complementary evidence. These effects support the complementary mechanisms described in Section 3.3.

Timing of route decisions. Table 5 compares five-seed mean accuracies when routes are fixed or selected in early, middle, and late blocks of the six-transition path. Fixing early routes reduces Portraits accuracy to 88.94, compared with 91.82 when only late routes are fixed. On Covertype, the corresponding values are 76.84 and 79.75. On Portraits, restricting adaptive selection to early transitions gives 90.05, compared with 86.47 when selection is restricted to late transitions. This pattern matches the propagation factors in Equation (S58): early decisions influence more subsequent states. With tolerance set to zero, Portraits and Covertype reach 90.84 and 78.06; the nominal stagewise tolerance reaches 93.35 and 82.02. Figure 3 summarizes the complementary library-size and tolerance interventions; Table S7 gives the complete tolerance comparison.

4.4 Computational behavior

PrismBridge reuses the three prediction streams across candidate mixtures. Its measured runtime is 1.04 times STDW on CIFAR-10 and is lower than both STDW and SWAT on Office-31. Table S11 reports all four benchmark timings, and Section S2.5 separates shared prediction from candidate optimization, including dependence on route count, views, and update budget.

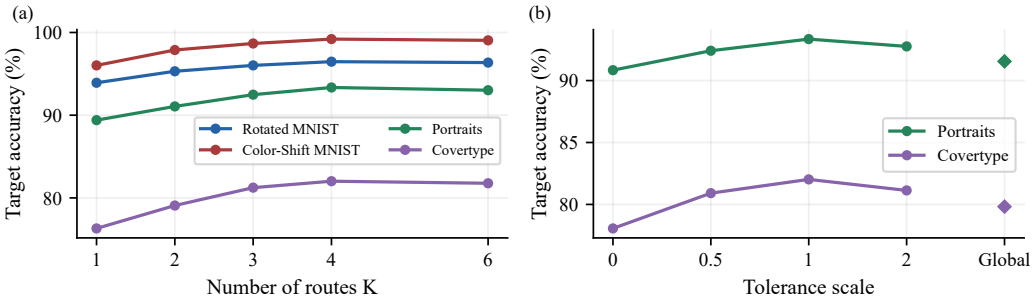


Figure 3: Accuracy versus (a) route-library size and (b) tolerance scale, including a global-tolerance control. Points are means over five seeds.

5 Related Work

Gradual adaptation and path construction. Analyses of gradual self-training connect the spacing of intermediate domains to cumulative target error (Kumar et al., 2020; Wang et al., 2022a; He et al., 2024). IDOL orders unindexed observations (Chen & Chao, 2021); GIFT and GOAT interpolate intermediate data (Abnar et al., 2021; He et al., 2024); gradient flows, normalizing flows, and semi-dual transport generate distribution paths (Zhuang et al., 2024; Sagawa & Hino, 2025; Chen et al., 2026). SWAT uses sliding-window adversarial alignment (Wang et al., 2025a), STDW schedules source and target loss weights (Wang et al., 2026), and OSMOSIS matches batches as the domains evolve (Wang et al., 2025b). PrismBridge selects a bridge rule at each supplied stage, making the supervision route an optimization variable and retaining the model trajectory induced by those choices.

Semantic anchors and prediction consistency. Source hypotheses retain class information learned before adaptation (Liang et al., 2020), prototypes summarize labeled class examples (Snell et al., 2017), and mixed-domain learning builds intermediate supervision (Na et al., 2021). Entropy regularization encourages confident predictions (Grandvalet & Bengio, 2004), while teacher consistency stabilizes predictions across perturbations (Tarvainen & Valpola, 2017). Confirmation-bias studies explain how repeated pseudo-label fitting can reinforce errors (Arazo et al., 2020; Chen et al., 2022a). Unsupervised model selection (Hu et al., 2024) and inter-class topology preservation (Ni et al., 2025) motivate evaluating candidate updates through both prediction quality and semantic structure. PrismBridge combines these signals to compare stage-specific bridges, with score tolerance and a predetermined preference determining which candidate supplies the next update.

6 Conclusion

Gradual domain adaptation requires effective decisions on how to utilize different sources of supervision as the learner moves along an evolving domain path. In this work, we introduce PrismBridge, which formulates bridge selection as an explicit route optimization problem rather than relying on a fixed adaptation rule or predefined schedule. By combining frozen source predictions, prototype-based semantic anchors, and current-model predictions, PrismBridge constructs adaptive bridges and selects suitable supervision routes through tolerance-aware optimization. Furthermore, our theoretical analysis connects stage-wise route decisions with their long-term effects on the induced model trajectory. The derived regret bounds and pathwise risk certificate provide a principled understanding of how route uncertainty, bridge quality, pseudo-label fitting, and domain changes influence adaptation performance. Extensive experiments across geometric, appearance, temporal, tabular, and corruption shifts demonstrate the effectiveness of PrismBridge. The consistent improvements over existing GDA and test-time adaptation approaches, together with component, route-size, tolerance, and early-decision analyses, validate the importance of adaptive route selection and its stabilization mechanisms. Overall, PrismBridge highlights the potential of treating supervision routing as an optimization problem for more flexible and reliable gradual domain adaptation.

REPRODUCIBILITY STATEMENT

The route state, losses, selector, and update sequence are specified in Sections 2–3 and Appendix S2. Appendix S3 gives the assumptions and complete proofs. Appendix S4 reports complementary corruption results, route interventions, path-length and resolution studies, sensitivity analyses, and runtime comparisons. Corruption accuracies are averaged uniformly across the 15 corruption types.

AI USE STATEMENT

Generative AI tools were used solely to assist with and polish the writing of this manuscript. The authors reviewed and revised all AI-assisted content and take full responsibility for the final manuscript.

REFERENCES

Samira Abnar, Rianne van den Berg, Golnaz Ghiasi, Mostafa Dehghani, Nal Kalchbrenner, and Hanie Sedghi. Gradual domain adaptation in the wild: When intermediate distributions are absent. *CoRR*, abs/2106.06080, 2021.

Eric Arazo, Diego Ortego, Paul Albert, Noel E. O'Connor, and Kevin McGuinness. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, 2020.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Mach. Learn.*, 79(1-2):151–175, 2010.

Jock A. Blackard and Denis J. Dean. Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables. *Computers and Electronics in Agriculture*, 24(3):131–151, 1999.

Baixu Chen, Junguang Jiang, Ximei Wang, Pengfei Wan, Jianmin Wang, and Mingsheng Long. Debiased self-training for semi-supervised learning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 32424–32437, 2022a.

Dian Chen, Dequan Wang, Trevor Darrell, and Sayna Ebrahimi. Contrastive test-time adaptation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 295–305, 2022b.

Hong-You Chen and Wei-Lun Chao. Gradual domain adaptation without indexed intermediate domains. *Advances in neural information processing systems*, 34:8201–8214, 2021.

Xinyang Chen, Sinan Wang, Mingsheng Long, and Jianmin Wang. Transferability vs. discriminability: Batch spectral penalization for adversarial domain adaptation. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pp. 1081–1090, 09–15 Jun 2019.

Zhichao Chen, Zhan Zhuang, Yunfei Teng, Hao Wang, Fangyikang Wang, Zhengnan Li, Tianqiao Liu, Haoxuan Li, and Zhouchen Lin. Rethinking the flow-based gradual domain adaptation: A semi-dual optimal transport perspective. *arXiv preprint arXiv:2602.01179*, 2026.

Nicolas Courty, Rémi Flamary, Devis Tuia, and Alain Rakotomamonjy. Optimal transport for domain adaptation. *IEEE transactions on pattern analysis and machine intelligence*, 39(9):1853–1865, 2016.

Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Edoardo Debenedetti, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. Robustbench: a standardized adversarial robustness benchmark. In *NeurIPS Datasets and Benchmarks*, 2021.

Yongxing Dai, Jun Liu, Yifan Sun, Zekun Tong, Chi Zhang, and Ling-Yu Duan. Idm: An intermediate domain module for domain adaptive person re-id. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11844–11854, 2021.

Bharath Bhushan Damodaran, Benjamin Kellenberger, Rémi Flamary, Devis Tuia, and Nicolas Courty. Deepjdot: Deep joint distribution optimal transport for unsupervised domain adaptation. In *ECCV (4)*, volume 11208, pp. 467–483, 2018.

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

Shiry Ginosar, Kate Rakelly, Sarah Sachs, Brian Yin, and Alexei A Efros. A century of portraits: A visual historical record of american high school yearbooks. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 1–7, 2015.

Rui Gong, Wen Li, Yuhua Chen, and Luc Van Gool. Dlow: Domain flow for adaptation and generalization. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2472–2481, 2019.

Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. *Advances in neural information processing systems*, 17, 2004.

Yifei He, Haoxiang Wang, Bo Li, and Han Zhao. Gradual domain adaptation: Theory and algorithms. *Journal of Machine Learning Research*, 25(361):1–40, 2024.

Dan Hendrycks and Thomas G. Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *ICLR (Poster)*, 2019.

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

Fredrik D. Johansson, David Sontag, and Rajesh Ranganath. Support and invertibility in domain-invariant representations. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89, pp. 527–536, 16–18 Apr 2019.

Matthieu Kirchmeyer, Alain Rakotomamonjy, Emmanuel de Bézenac, and Patrick Gallinari. Mapping conditional distributions for domain adaptation under generalized target shift. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*, 2022.

Ananya Kumar, Tengyu Ma, and Percy Liang. Understanding self-training for gradual domain adaptation. In *International conference on machine learning*, pp. 5468–5479, 2020.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Mengxue Li, Yi-Ming Zhai, You-Wei Luo, Peng-Fei Ge, and Chuan-Xian Ren. Enhanced transport distance for unsupervised domain adaptation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13933–13941, 2020.

Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *International conference on machine learning*, pp. 6028–6039, 2020.

Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pp. 97–105, 2015.

Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I. Jordan. Deep transfer learning with joint adaptation networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pp. 2208–2217, 06–11 Aug 2017.

Mingsheng Long, ZHANGJIE CAO, Jianmin Wang, and Michael Jordan. Conditional adversarial domain adaptation. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Robert A Marsden, Mario Döbler, and Bin Yang. Introducing intermediate domains for effective self-training during test-time. In *2024 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–10, 2024.

Jaemin Na, Heechul Jung, Hyung Jin Chang, and Wonjun Hwang. Fixbi: Bridging domain spaces for unsupervised domain adaptation. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1094–1103, 2021.

Chenggong Ni, Fan Lyu, Jiayao Tan, Fuyuan Hu, Rui Yao, and Tao Zhou. Maintaining consistent inter-class topology in continual test-time adaptation. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15319–15328, 2025.

Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

Sinno Jialin Pan, Ivor W. Tsang, James T. Kwok, and Qiang Yang. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 22(2):199–210, 2011.

Shogo Sagawa and Hideitsu Hino. Gradual domain adaptation via normalizing flows. *Neural Comput.*, 37(3):522–568, 2025.

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

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017.

Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *European conference on computer vision*, pp. 443–450, 2016.

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

Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *CoRR*, abs/1412.3474, 2014.

Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2962–2971, 2017.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno A. Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *ICLR*, 2021.

Haoxiang Wang, Bo Li, and Han Zhao. Understanding gradual domain adaptation: Improved analysis, optimal path and beyond. In *International Conference on Machine Learning*, pp. 22784–22801, 2022a.

Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai. Continual test-time domain adaptation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7191–7201, 2022b.

Zixi Wang, Yubo Huang, Wenwei Luo, Tonglan Xie, Mengmeng Jing, and Lin Zuo. SWAT: sliding window adversarial training for gradual domain adaptation. *CoRR*, abs/2501.19155, 2025a.

Zixi Wang, Yubo Huang, Jingzehua Xu, Jinzhu Wei, Shuai Zhang, and Xin Lai. Multi-modal gradual domain osmosis: Stepwise dynamic learning with batch matching for gradual domain adaptation. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 8959–8967, New York, NY, USA, 2025b.

Zixi Wang, Yushe Cao, Yubo Huang, Jinzhu Wei, Jingzehua Xu, Shuai Zhang, and Xin Lai. Self-training with dynamic weighting for robust gradual domain adaptation. *Advances in Neural Information Processing Systems*, 38:107257–107278, 2026.

Han Zhao, Remi Tachet Des Combes, Kun Zhang, and Geoffrey Gordon. On learning invariant representations for domain adaptation. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pp. 7523–7532, 09–15 Jun 2019.

Zhan Zhuang, Yu Zhang, and Ying Wei. Gradual domain adaptation via gradient flow. In *International Conference on Learning Representations*, volume 2024, pp. 26953–26978, 2024.

APPENDIX CONTENTS

S1 Notation **14**

S2 Implementation details **14**

S2.1 Route-library construction and parameter selection 14

S2.2 Masks, centroids, and exact losses 15

S2.3 Candidate evaluation and deterministic selection 16

S2.4 Hyperparameter sensitivity and architectures 16

S2.5 Computational cost and memory accounting 17

S3 Proofs and supplementary analysis **17**

S3.1 Route relaxation and the fixed-route constraint 17

S3.2 Sequential optimality and trajectory regret 19

S3.3 Tolerance-aware route selection 21

S3.4 Prototype bridging and confidence filtering 23

S3.5 Supervised smoothing and semantic stability 25

S3.6 Target-risk guarantees and path perturbations 26

S4 Detailed experimental evidence **28**

S4.1 Benchmark protocols and metrics 28

S4.2 Severity-5 comparison protocols and aggregation 29

S4.3 Isolating route size, tolerance, and path position 29

S4.4 Route-policy isolation with identical stabilizers 30

S4.5 Path density and hyperparameter sensitivity 30

S4.6 Effects of path length and image resolution 31

S4.7 Measured runtime 33

S1 NOTATION

Domain indices run from 0 through T; stage t updates the learner using B_{t+1}. A route is an update configuration, while a policy chooses routes across stages. Table S1 collects the principal symbols.

Table S1: Principal symbols. State superscripts distinguish trajectories induced by different policies.

Symbol	Meaning
$\mathcal{D}_t, P_t, B_t$	Joint domain distribution, input marginal, and observed sample
T, C, d	Number of transitions, classes, and feature dimensions
h_t, f_t, s_t	Classifier, encoder, and complete adaptation state
$r, \mathcal{R}, K$	Route tuple, route library, and its cardinality
π, s_t^π	Route policy and its induced state after stage $t-1$
$\mathcal{G}_t^r$	Complete update from stage t to stage $t+1$
$c_t, J(\pi)$	Transition certificate and cumulative path objective
V_t^*, Q_t^*	Optimal remaining cost and route cost-to-go
q_s, q_p, q_t	Frozen source, prototype, and current prediction streams
$w_r, \widetilde{q}_r$	Route mixture weights and mixed bridge probabilities
$\mathcal{V}_r, V$	View set and number of evaluated views
m_{ri}, η, κ_r	Retention mask, confidence threshold, agreement rule
$S_t, \delta_t, \prec_t$	Maximized route score, tolerance, configured preference order
ϵ_t, Ω_t	Score calibration error and continuation range
D_t, b_t, a_t, ρ_t	Domain change, bridge displacement, fitting KL, coverage

Among the quantities in Table S1, theoretical optimal values carry an asterisk on J, V, Q; the selected route r_t^* denotes the output of the tolerance-aware decision rule. The simplex for K route probabilities is Δ^{K-1}. Losses are minimized; scores are maximized; the first eligible route is taken in the configured preference order. This convention is used throughout the manuscript.

S2 IMPLEMENTATION DETAILS

S2.1 ROUTE-LIBRARY CONSTRUCTION AND PARAMETER SELECTION

The library contains bridge-view-consensus tuples of the form defined in Section 3.3. Table S2 makes the four route labels explicit. The dominant-weight conditions describe semantic emphases; a concrete route is identified by its full tuple of weights, views, and consensus parameters. Changing the selected tuple in Table S2 adapts the update rule while preserving the supplied domain order.

Table S2: Meaning of route labels in Figure 2. All rows satisfy $\alpha_r, \beta_r, \gamma_r \geq 0$ and $\alpha_r + \beta_r + \gamma_r = 1$.

Route	Weight interpretation	Semantic role
Source-anchored	$\alpha_r \geq \max(\beta_r, \gamma_r)$	Frozen classifier reference
Prototype-guided	$\beta_r \geq \max(\alpha_r, \gamma_r)$	Class-centroid geometry
Target-driven	$\gamma_r \geq \max(\alpha_r, \beta_r)$	Adapted current predictions
Mixed	$\alpha_r, \beta_r, \gamma_r > 0$	Combined prediction streams

The configuration separates three sources of parameter choices. Fixed values specify route mixtures, library size, views, loss coefficients, optimizer, adaptation budget, and the deterministic preference order before adaptation. Source-selected values use a held-out labeled source split when source-side selection is required. Stage-adaptive quantities, including confidence quantiles, masks, route scores, and tolerances, are computed from the current unlabeled stage. Route construction, confidence thresholds, and stopping decisions follow the fixed configuration, labeled source validation, and unlabeled agreement statistics. The sensitivity studies characterize how the resulting accuracy varies with each parameter. Table S3 records the evaluated ranges, and Tables S6 and S7 record the route-count and tolerance interventions.

The candidate construction has three complementary axes: anchor mixtures vary $(\alpha_r, \beta_r, \gamma_r)$, view variants vary $\mathcal{V}_r$, and consensus variants vary the acceptance policy κ_r. Thus $\mathcal{R} = \mathcal{R}^{\text{anchor}} \cup \mathcal{R}^{\text{view}} \cup \mathcal{R}^{\text{cons}}$, with identical full tuples included only once. These families can overlap: a prototype-guided route can also use multiple views and a consensus filter. The four semantic labels in Table S2 describe mixture emphasis; the three construction axes describe how candidate tuples differ. The stage index denotes evaluation at the current state, so library entries retain their identities across transitions.

S2.2 Masks, centroids, and exact losses

For a route r, compute bridge probabilities on each prescribed view $a_v(x_i)$ and let $\widetilde{y}_{ri}$ be the class predicted by the reference-view bridge. An explicit confidence-and-consensus mask is

$$m_{ri} = \mathbf{1}\{\max_c \widetilde{q}_r(c \mid x_i) \geq \eta\}\, \mathbf{1}\left\{ \frac{1}{V} \sum_{v=1}^{V} \mathbf{1}\{\arg\max_c \widetilde{q}_r(c \mid a_v(x_i)) = \widetilde{y}_{ri}\} \geq a_r \right\}. \tag{S1}$$

Here $\eta \in [0,1]$ is the confidence threshold and $a_r \in [0,1]$ is the required agreement fraction. They are acceptance parameters specified by the policy κ_r, which also records the confidence-filter mode and fallback choice. Class-index ties use the smallest class index. Every route has at least one view; label preservation is required of each augmentation. The identity view may be included. A consensus fallback can replace a disputed reference-view label by the class of the mean bridge distribution before the same acceptance tests; the choice of fallback belongs to the policy specification. Source prototypes are formed from a labeled reference sample containing every class; the cached prototypes then provide the class references for subsequent replay minibatches.

Class-conditional confidence filtering. A quantile-based policy replaces the first indicator in Equation (S1) by a class-wise retention test. Let $g_{ri} = \max_c \widetilde{q}_r(c \mid x_i)$ and $\mathcal{B}_{r,c} = \{i : \widetilde{y}_{ri} = c\}$. Let $\text{rank}_{r,c}(i)$ order this class by decreasing g_{ri}, breaking equal-confidence ties by increasing sample index. For a prescribed fraction $\chi_{r,c} \in [0,1]$, define

$$k_{r,c} = \lceil \chi_{r,c} |\mathcal{B}_{r,c}| \rceil, \qquad m_{ri}^{\text{conf}} = \mathbf{1}\{\text{rank}_{r,\widetilde{y}_{ri}}(i) \leq k_{r,\widetilde{y}_{ri}}\}. \tag{S2}$$

For $k_{r,c} > 0$, the confidence at rank $k_{r,c}$ is the empirical threshold: retain all entries strictly above it and only the required number of tied entries at it. Thus exactly $k_{r,c}$ entries survive, including when confidences coincide. The endpoints retain none at $\chi_{r,c} = 0$ and all at $\chi_{r,c} = 1$; an empty class contributes no entries. The final mask multiplies m_{ri}^{conf} by the agreement indicator in Equation (S1), so the final coverage can be smaller than the pre-consensus quota. Both fixed thresholds and class-wise quantiles use unlabeled predictions. The configuration determines the filter mode and fractions; the fractions are distinct from the agreement requirement a_r. Appendix S3.4 derives the quantile rule from a coverage-constrained optimization.

Let $z_i(\theta) = f_\theta(x_i) / \max(\|f_\theta(x_i)\|_2, \varepsilon)$, where $\varepsilon > 0$ is a numerical normalization constant. Set $u_{ri,c} = m_{ri}\, \text{sg}[\widetilde{q}_r(c \mid x_i)]$ and $n_{r,c} = \sum_i u_{ri,c}$. For supported classes, the normalized target centroid is

$$p_c^r(\theta) = \frac{\sum_i u_{ri,c} z_i(\theta)}{n_{r,c}}, \qquad \bar{p}_c^r(\theta) = \frac{p_c^r(\theta)}{\max(\|p_c^r(\theta)\|_2, \varepsilon)}. \tag{S3}$$

Define $\mathcal{P}_r = \{(c,d) : c < d,\ n_{r,c} > 0,\ n_{r,d} > 0\}$. Source normalized centroids and their squared-distance matrix are computed analogously from true source class memberships. The topology loss in Equation (8) compares scalar class-pair distances, linking source and current class geometry. The prototype stream in Equation (5) measures sample-to-prototype distances within the frozen source feature space.

Equations (7)–(9) contain all optimization losses, with nonnegative coefficients and $\tau_t \geq 0$. The bridge objective combines hard pseudo-label fitting, soft bridge alignment, and inter-class geometry; the smoothing objective combines true source supervision, retained target supervision, and multi-view agreement. Bridge distributions and masks are detached during each candidate update. They may be recomputed after a stage is committed. In route-specific losses, $V = |\mathcal{V}_r|$. The multi-view loss is averaged over retained samples and views, and its evaluation value supplies the agreement penalty in Equation (10). The coefficient λ_v controls agreement during smoothing, whereas τ_t additionally weights agreement in route scoring.

If $\sum_i m_{ri} = 0$, set $M_r = 1$, retain zero target contributions, and execute the source-supervised part of the update. If fewer than two classes have retained support, $\mathcal{P}_r$ is empty and its penalty is zero. These conventions define the empirical losses and tie decisions for low-support stages.

S2.3 Candidate evaluation and deterministic selection

All routes share the stage batches, view realizations, and initial state s_t, including its model and optimizer components. Their bridge targets use the initial classifier h_t, frozen source model h_0, and prototypes P^s. After H_b bridge steps, each route is scored at its updated candidate state s_t^r.

The score-tolerance set contains the eligible routes, and the configured strict total order $\prec_t$ determines the first eligible member. Applying H_s source-supervised smoothing steps to this candidate yields the next-stage state s_{t+1}. The configuration fixes $\prec_t$ before adaptation.

Algorithm S1 expands one stage of Algorithm 1. The operator $\mathrm{Filter}(\widetilde{q}_r, \kappa_r)$ produces labels and masks by the rules in Section S2.2. The operator $\mathrm{Optimize}(s, L, H)$ applies H configured optimizer steps to L from s and returns the updated state; it returns s when $H = 0$. Bridge targets, labels, and masks remain fixed during both optimization phases, as do the source model and prototypes. The notation $\widehat{L}(s_t^r)$ evaluates a loss at the candidate model on the shared evaluation inputs.

Algorithm S1 PrismBridge at one adaptation stage

Input: State s_t; source and target batches B_0, B_{t+1}; frozen h_0, P^s; route library $\mathcal{R}$; budgets H_b, H_s; tolerance δ_t; preference order $\prec_t$.
Output: Next state s_{t+1} and selected route r_t^*.

Compute q_s, q_p, q_t on the target inputs and their prescribed views.
for $r \in \mathcal{R}$ **do**
  $s_t^r \leftarrow s_t$.
  $\widetilde{q}_r \leftarrow \mathrm{sg}[\alpha_r q_s + \beta_r q_p + \gamma_r q_t]$.
  $(\widetilde{y}_r, m_r) \leftarrow \mathrm{Filter}(\widetilde{q}_r, \kappa_r)$.
  $s_t^r \leftarrow \mathrm{Optimize}(s_t^r, L^r_{\mathrm{bridge}}, H_b)$ using Equation (8).
  $S_t(r) \leftarrow -\widehat{L}^r_{\mathrm{bridge}}(s_t^r) - \widehat{L}^r_{\mathrm{smooth}}(s_t^r) - \tau_t \widehat{L}^r_{\mathrm{con}}(s_t^r)$.
end for
$S_t^{\max} \leftarrow \max_{r \in \mathcal{R}} S_t(r)$.
$\mathcal{T}_t(\delta_t) \leftarrow \{r \in \mathcal{R} : S_t(r) \geq S_t^{\max} - \delta_t\}$.
$r_t^* \leftarrow \mathrm{first}_{\prec_t} \mathcal{T}_t(\delta_t)$.
$s_{t+1} \leftarrow \mathrm{Optimize}(s_t^{r_t^*}, L^{r_t^*}_{\mathrm{smooth}}, H_s)$ using Equation (9).
return (s_{t+1}, r_t^*).

Only the selected state in Algorithm S1 initializes the next stage. The state-retaining version stores one completed state per candidate. Alternatively, sequential evaluation stores only scores and replay seeds, and replays the selected trial from the cached state after the tolerance set is known. Reusing the same inputs and random stream with deterministic update operations reproduces that state. This option trades an additional H_b optimization steps for reduced storage by reconstructing the selected trial on demand. Section S2.5 accounts for both versions.

In Equation (11), maximizing S_t determines the eligibility threshold, while $\prec_t$ selects the first admissible candidate. Equivalently, the tolerance set satisfies $-S_t(r) \leq \min_j[-S_t(j)] + \delta_t$. The utility and cost forms give the same eligible set and the same first-eligible route under the configured preference order.

S2.4 Hyperparameter sensitivity and architectures

The MNIST variants and lower-resolution Portraits use a three-layer 64-channel CNN with a 1024-unit MLP head. The MNIST configuration has one input channel and the Portraits CNN has three. Covertype uses a 256-unit fully connected encoder and a 256-unit MLP head. Expanded Portraits uses ResNet-18 with a 1024-unit MLP head, and Office-31 uses ResNet-50 with a 1024-unit MLP head. The corruption experiments use a RobustBench source model (Croce et al., 2021). Architecture

Table S3: Hyperparameter values evaluated in the sensitivity studies.

Parameter	Evaluated values	Analysis
Source weight λ_s	$0.1, 0.3, 0.5, 1.0, 2.0$	Portraits sensitivity
Topology weight λ_g	$0, 0.01, 0.05, 0.1, 0.2$	Portraits sensitivity
Stability weight τ_t	$0, 0.1, 0.3, 0.5, 1.0$	Portraits sensitivity
Batch size	$32, 64, 128, 256$	Portraits sensitivity
Route count K	$1, 2, 3, 4, 6$	Four ordered benchmarks
Tolerance multiplier	$0, 0.5, 1, 2$; global	Portraits, Covertype

and path resolution define the protocols detailed in Appendix S4. Optimization uses Adam, weight decay 10^{-4}, and gradient clipping at 5.0 on RTX 5090 hardware.

S2.5 Computational cost and memory accounting

Let B denote batch size, C class count, d feature dimension, K route count, V the number of distinct cached views across the library, H_b candidate optimization steps, and H_s selected-route smoothing steps. Thus each route uses at most V views. Let F be a student forward-pass cost, F_0 a frozen source forward-pass cost, and U the cost of one candidate optimization step including its required backward pass. Three-stream prediction and mixture construction have the schematic cost

$$O\big(V(F_0 + F) + VBCd + KVBC\big). \tag{S4}$$

The $VBCd$ term accounts for distances to source prototypes; $KVBC$ accounts for mixtures and per-sample confidence and agreement operations. The encoder outputs can be shared across routes at the same pre-update state. Equation (6) constructs all K bridge distributions by mixing these shared predictions. Evaluating topology additionally costs $O(BCd)$ for weighted centroids and $O(C^2d)$ for class-pair distances per candidate, hence $O(K(BCd + C^2d))$ across the library. Sorting class-wise confidences, when used, adds at most $O(KB \log B)$; forming the tolerance set and scanning the configured preference order is linear in K once diagnostics are available.

If every candidate is trial-trained, the total per-stage cost additionally contains

$$KH_bU + KVF_{\text{eval}} + H_sU, \tag{S5}$$

where F_{eval} is the forward cost of evaluating a trial model. These trial evaluations require candidate-specific encoder passes. Replaying the selected trial adds H_bU to Equation (S5). With bridge-only scoring, $H_b = 0$ and the evaluation terms are shared. Thus per-stage computation depends jointly on the number of routes, the number of views, and the optimization budget.

Let M_θ denote model and optimizer storage, M_{act} the peak activation storage of one trial, and M_{input} the cached batches and views. Sequential evaluation with selected-trial replay can use

$$O\big(M_\theta + M_{\text{act}} + M_{\text{input}} + VBC + Cd + K\big) \tag{S6}$$

up to the constant number of source, starting-state, and selected-state buffers. Candidate targets and centroids are constructed one route at a time from shared predictions. Retaining all completed candidate states instead requires $O(KM_\theta)$ state storage; retaining their computation graphs additionally requires $O(KM_{\text{act}})$. This accounting complements the measured wall-clock comparisons in Table S11.

S3 Proofs and supplementary analysis

S3.1 Route relaxation and the fixed-route constraint

Fix the domain samples, the source initialization, the optimization budgets, and the random streams used by the candidate update operators. Assume $K, T \geq 1$, finite stage costs, and measurable states and update operators. Including random seeds and optimizer variables in the state makes Equation (1) deterministic conditional on these quantities. Two policies starting from the same state can produce

different states after their first different route choice. In particular, evaluating a later route requires the state generated by the full route prefix.

For a route sequence $\mathbf{r} = (r_0, \dots, r_{T-1})$, write

$$s_t(\mathbf{r}_{<t}) = \mathcal{G}_{t-1}^{r_{t-1}} \circ \cdots \circ \mathcal{G}_0^{r_0}(s_0), \qquad F(\mathbf{r}) = \sum_{t=0}^{T-1} c_t(s_t(\mathbf{r}_{<t}), r_t). \tag{S7}$$

The composition is empty at $t = 0$. The current sample and source replay arguments of each operator are suppressed. This explicit notation identifies where path dependence enters: changing r_j changes every s_t with $t > j$, and hence may change every subsequent summand of F.

Proof of Proposition 3.1. For each k, the policy π^k belongs to Π. The infimum over Π therefore cannot exceed the value of this particular policy. Applying this argument to every k gives the chain

$$\begin{aligned} \inf_{\pi\in\Pi} \left[R_0(h_0) + \sum_t c_t(s_t^\pi, \pi_t(s_t^\pi)) \right] \le R_0(h_0) + \sum_t c_t(s_t^{\pi^k}, k) \quad \text{for every } k, \\ \implies \quad J^*_{\text{relax}} \le \min_k J(\pi^k) = J^*_{\text{fixed}}. \end{aligned} \tag{S8}$$

The comparison follows from embedding the fixed-policy set into the adaptive-policy set, with every policy retaining its own induced trajectory and stage costs.

For the conditional statement, set $a_{tk} = c_t(\bar{s}_t, k)$ and $a_t^{\min} = \min_k a_{tk}$; these are now fixed numbers. Every $z_t \in \Delta^{K-1}$ satisfies

$$\sum_k z_{t,k} a_{tk} \ge \sum_k z_{t,k} a_t^{\min} = a_t^{\min}. \tag{S9}$$

Choosing a vertex $z_t = e_{k_t}$ with $k_t \in \arg\min_k a_{tk}$ attains equality. Because the feasible set is a Cartesian product over stages, these minimizing vertices can be chosen independently. Hence the first equality in Equation (3) follows. Finally, $\sum_t a_t^{\min} \le \sum_t a_{tk}$ for every fixed k; taking the minimum over those upper bounds proves the second inequality. □

The conditional relaxation is tight with respect to a fixed route precisely when at least one route minimizes all stages simultaneously:

$$\sum_t \min_k a_{tk} = \min_k \sum_t a_{tk} \quad \iff \quad \bigcap_{t=0}^{T-1} \arg\min_k a_{tk} \ne \varnothing. \tag{S10}$$

To verify the reverse implication, choose a common minimizer. For the forward implication, let k^* minimize the right-hand side. The nonnegative terms $a_{t,k^*} - a_t^{\min}$ sum to zero, so each term is zero. The equality condition characterizes route mismatch for the conditional fixed-state problem: a common minimizing route makes the constraint tight, whereas different stagewise minimizers create a relaxation gap.

A two-stage example that separates local and global decisions. Consider two initial routes A, B with costs $c_0(s_0, A) = 0$ and $c_0(s_0, B) = 1$. Route A produces state a and route B produces state b. At the last stage, suppose $c_1(a, A) = c_1(a, B) = 10$ and $c_1(b, A) = c_1(b, B) = 0$. Greedy local selection chooses A and has total cost 10; the optimal policy chooses B and has total cost 1. All costs are nonnegative and both routes are available at both stages. The example shows how continuation costs determine the best initial route and, consequently, the cost of the complete adaptation path.

Switching-penalty interpolation. The fixed-route constraint can also be exposed by penalizing changes between consecutive decisions. For the trajectory cost F in Equation (S7), let $N(\mathbf{r}) = \sum_{t=1}^{T-1} \mathbf{1}\{r_t \ne r_{t-1}\}$ and define

$$J^*_\lambda = R_0(h_0) + \min_{\mathbf{r}\in\mathcal{R}^T} \{F(\mathbf{r}) + \lambda N(\mathbf{r})\}, \qquad \lambda \ge 0. \tag{S11}$$

Every $F(\mathbf{r})$ is evaluated on that sequence's own states. Conditional on the fixed random streams, a policy induces a sequence and every sequence is implementable by a policy, so $J^*_0 = J^*_{\text{relax}}$. Constant sequences have $N = 0$, which also gives

$$J^*_{\text{relax}} = J^*_0 \le J^*_\lambda \le J^*_{\lambda'} \le J^*_{\text{fixed}}, \qquad 0 \le \lambda \le \lambda'. \tag{S12}$$

The middle inequality follows pointwise before minimization: increasing a nonnegative penalty cannot decrease the cost of any sequence.

Let $F_{\min} = \min_{\mathbf{r}} F(\mathbf{r})$ and $F_{\text{fix}} = \min_k F(k, \dots, k)$. Any nonconstant sequence has at least one switch, hence

$$F(\mathbf{r}) + \lambda N(\mathbf{r}) \geq F_{\min} + \lambda > F_{\text{fix}} \quad \text{if } \lambda > F_{\text{fix}} - F_{\min}. \tag{S13}$$

Consequently every minimizer is constant beyond this finite threshold and $J^*_\lambda = J^*_{\text{fixed}}$. For a minimizing sequence $\mathbf{r}^\lambda$ at any $\lambda > 0$, comparison with the best constant sequence further yields $\lambda N(\mathbf{r}^\lambda) \leq F_{\text{fix}} - F_{\min}$. The switching penalty interpolates between adaptive and fixed policies while evaluating each sequence on its induced states. This family of comparison programs complements the score-tolerance selector in Equation (11), whose decisions follow the configured route preference.

Conditional simplex and continuous-path forms. For the fixed-state costs $a_{tk} = c_t(\bar{s}_t, k)$ used in Equation (3), the corresponding convex program is

$$A_\lambda = \min_{z_t \in \Delta^{K-1}} \left\{ \sum_t \langle a_t, z_t \rangle + \lambda \sum_{t=1}^{T-1} \|z_t - z_{t-1}\|_1 \right\}. \tag{S14}$$

At $\lambda = 0$ this is the conditional envelope. Write $D(z) = \sum_t \langle a_t, z_t \rangle$, $D_{\min} = \sum_t \min_k a_{tk}$, and $D_{\text{fix}} = \min_k \sum_t a_{tk}$. Comparison with a constant minimizing vertex gives $D(z^\lambda) + \lambda \operatorname{Var}_1(z^\lambda) \leq D_{\text{fix}}$, hence $\operatorname{Var}_1(z^\lambda) \leq (D_{\text{fix}} - D_{\min})/\lambda$. For any diverging sequence of penalties, compactness gives a subsequence of minimizers converging to a constant $\bar{z}$. Its data term satisfies $D(\bar{z}) = \sum_k \bar{z}_k \sum_t a_{tk} \geq D_{\text{fix}}$. Since $A_\lambda \geq D(z^\lambda)$ and $A_\lambda \leq D_{\text{fix}}$, continuity squeezes this subsequence's values to D_{fix}. Monotonicity of A_λ proves convergence of the full family. These statements apply to conditional fixed-state costs; Equation (S11) supplies the comparison for evolving states.

To obtain an exact path representation, set $z(u) = z_t$ and $\psi(u) = T a_t$ on $u \in [t/T, (t+1)/T)$. Define $\operatorname{Var}_1(z)$ as the sum of the ℓ_1 magnitudes of its interior jumps. Then

$$\int_0^1 \langle \psi(u), z(u) \rangle \, du + \lambda \operatorname{Var}_1(z) = \sum_t \langle a_t, z_t \rangle + \lambda \sum_{t=1}^{T-1} \|z_t - z_{t-1}\|_1. \tag{S15}$$

The factor T compensates for interval length $1/T$. For an absolutely continuous selector, the variation is instead $\int_0^1 \|\dot{z}(u)\|_1 \, du$. Both forms express the same distinction between stage costs and temporal resistance to changing the route.

The nondifferentiable penalty and simplex boundary are handled explicitly. At an optimum of Equation (S14), there exist $\xi_t \in \partial \|z_t - z_{t-1}\|_1$ for $1 \leq t < T$, with $\xi_0 = \xi_T = 0$, such that

$$0 \in a_t + \lambda(\xi_t - \xi_{t+1}) + N_{\Delta^{K-1}}(z_t), \qquad 0 \leq t < T. \tag{S16}$$

Here $N_\Delta(z) = \{v : \langle v, u - z \rangle \leq 0 \; \forall u \in \Delta\}$ is the normal cone, and each coordinate of ξ_t equals the sign of the corresponding nonzero difference or belongs to $[-1, 1]$ at zero. The difference $\xi_t - \xi_{t+1}$ comes from the two adjacent variation terms. Convexity makes this subgradient condition sufficient as well as necessary, characterizing the solution in both the interior and the boundary of the simplex.

S3.2 Sequential optimality and trajectory regret

For a state s at stage t, define the optimal remaining cost by

$$V_t^*(s) = \min_{r_t, \dots, r_{T-1}} \sum_{j=t}^{T-1} c_j(s_j, r_j), \quad s_t = s, \quad s_{j+1} = \mathcal{G}_j^{r_j}(s_j). \tag{S17}$$

Because the route set and horizon are finite, the minimum over the K^{T-t} sequences exists whenever stage costs are finite. Splitting the first route from the continuation yields

$$\begin{aligned} V_t^*(s) &= \min_{r \in \mathcal{R}} \left[c_t(s, r) + \min_{r_{t+1:T-1}} \sum_{j=t+1}^{T-1} c_j(s_j, r_j) \right] \\ &= \min_{r \in \mathcal{R}} \left[c_t(s, r) + V_{t+1}^*(\mathcal{G}_t^r(s)) \right]. \end{aligned} \tag{S18}$$

The inner minimization starts at the state generated by r. This establishes Equation (4): the optimal stage decision minimizes Q_t^*, combining its immediate cost with the optimal continuation from the resulting state. The full value is $J^*_{\text{relax}} = R_0(h_0) + V_0^*(s_0)$.

The same argument extends to a simplex-valued distribution over route decisions. Suppose $z_t(\cdot \mid s) \in \Delta^{K-1}$ samples one route, executes that route's update, and then uses an optimal continuation. The conditional expected cost is

$$\sum_{r=1}^{K} z_t(r \mid s) \left[c_t(s, r) + V_{t+1}^*(\mathcal{G}_t^r(s)) \right]. \tag{S19}$$

It is linear in z_t at the current state. Its minimum is attained at a vertex corresponding to a minimizer of $Q_t^*(s, \cdot)$. Backward induction therefore proves that optimizing over randomized policies gives the same optimal value as optimizing over deterministic policies. The simplex representation therefore preserves the adaptive optimum, with each route evaluated through its own continuation value $V_{t+1}^*(\mathcal{G}_t^r(s))$.

For stochastic operators, use nonanticipating policies and a state containing the available history. With integrable costs and measurable transition kernels, replace the continuation by $\mathbb{E}[V_{t+1}^*(s') \mid s, r]$. The minimization remains linear in a distribution over routes; choosing the smallest minimizing index gives a measurable deterministic selector for finite K. The policy-inclusion argument also holds for expected costs. This stochastic recursion averages future randomness before choosing a route, whereas the preceding deterministic analysis conditions on the random streams.

Tolerance and trajectory regret. Fix a stage and its realized state. Set $\widehat{c}(r) = -S_t(r)$, $\widehat{c}_{\min} = \min_r \widehat{c}(r)$, and write $\widehat{r} = r_t^*$. Equation (11) is exactly the requirement $\widehat{c}(\widehat{r}) \le \widehat{c}_{\min} + \delta_t$, followed by taking the first eligible route under $\prec_t$. Thus the local cost argument applies to any choice inside the tolerance set, while the configured preference order determines which admissible route is actually used.

Proof of Theorem 3.1. Let $r^c \in \arg\min_r c_t(s_t, r)$. Uniform score calibration gives

$$\begin{aligned} c_t(s_t, \widehat{r}) - c_t(s_t, r^c) &= \left[c_t(s_t, \widehat{r}) - \widehat{c}_t(s_t, \widehat{r}) \right] + \left[\widehat{c}_t(s_t, \widehat{r}) - \widehat{c}_t(s_t, r^c) \right] + \left[\widehat{c}_t(s_t, r^c) - c_t(s_t, r^c) \right] \\ &\le \epsilon_t + \delta_t + \epsilon_t. \end{aligned} \tag{S20}$$

The middle bound uses $\widehat{c}_{\min} \le \widehat{c}_t(s_t, r^c)$, so r^c need not minimize the observed score. This proves Equation (12).

For the cumulative statement, abbreviate $s_t = s_t^{\widehat{\pi}}$ and define the Bellman residual

$$A_t = c_t(s_t, \widehat{r}_t) + V_{t+1}^*(s_{t+1}) - V_t^*(s_t) \ge 0. \tag{S21}$$

Nonnegativity follows from the Bellman minimum at the very same state s_t. Since $V_T^* = 0$, summing these residuals produces the exact identity

$$\begin{aligned} \sum_{t=0}^{T-1} A_t &= \sum_t c_t(s_t, \widehat{r}_t) + \sum_t \left[V_{t+1}^*(s_{t+1}) - V_t^*(s_t) \right] \\ &= \sum_t c_t(s_t, \widehat{r}_t) - V_0^*(s_0) = J(\widehat{\pi}) - J^*_{\text{relax}}. \end{aligned} \tag{S22}$$

The telescoping follows the selected trajectory; at each visited state, V_t^* supplies the optimal remaining cost used in the comparison.

Let r_t^Q minimize $Q_t^*(s_t, \cdot)$ and abbreviate $v_t(r) = V_{t+1}^*(\mathcal{G}_t^r(s_t))$. Then

$$\begin{aligned} A_t &= c_t(s_t, \widehat{r}_t) - c_t(s_t, r_t^Q) + v_t(\widehat{r}_t) - v_t(r_t^Q) \\ &\le c_t(s_t, \widehat{r}_t) - \min_r c_t(s_t, r) + \max_r v_t(r) - \min_r v_t(r) \\ &\le \delta_t + 2\epsilon_t + \Omega_t(s_t). \end{aligned} \tag{S23}$$

Combining Equations (S22) and (S23) proves Equation (13). If the calibration assumption instead holds for Q_t^*, apply Equation (S20) directly to Q_t^*; then $A_t \le \delta_t + 2\epsilon_t$ and no continuation-range term is needed. □

The theorem separates three sources of approximation. The tolerance is an explicit choice made by the algorithm. Calibration measures the discrepancy between the chosen score and the theoretical cost. The continuation range measures the consequences of local selection for later transitions. Their sum gives the stagewise contribution to the trajectory regret.

S3.3 TOLERANCE-AWARE ROUTE SELECTION

Fix a stage, its realized state, and its finite candidate evaluations. Let $\sigma(1) \prec_t \cdots \prec_t \sigma(K)$ list the configured preference, let $M = \max_r S_t(r)$, and define the score gap $d_r = M - S_t(r) \geq 0$. Because a score maximizer has gap zero, every nonnegative tolerance has a nonempty eligible set. The selector can therefore be written exactly as

$$k(\delta) = \min\{j : d_{\sigma(j)} \leq \delta\}, \qquad r(\delta) = \sigma(k(\delta)), \qquad 0 \leq M - S_t(r(\delta)) \leq \delta. \tag{S24}$$

The last inequality follows from eligibility itself. It bounds the score difference admitted by the configured preference within the tolerance set.

Proposition S3.1 (Preference thresholds and monotonicity)**.** *With the preceding fixed evaluations and order, route $\sigma(k)$ is selected exactly for the tolerances*

$$d_{\sigma(k)} \leq \delta < \min_{j<k} d_{\sigma(j)}, \qquad \min \varnothing = +\infty. \tag{S25}$$

Moreover, for $0 \leq \delta \leq \delta'$, the selected preference ranks satisfy

$$k(\delta') \leq k(\delta). \tag{S26}$$

Proof. Selecting $\sigma(k)$ requires that it be eligible, which is precisely $d_{\sigma(k)} \leq \delta$. It also requires that every earlier entry fail eligibility, or $d_{\sigma(j)} > \delta$ for all $j < k$. Conjoining those conditions gives Equation (S25). Conversely, these inequalities admit entry k and exclude every earlier entry, so the first-eligible scan returns exactly $\sigma(k)$. This proves both directions, including score ties: a later entry tied in score with an earlier one has an empty selection interval when that earlier entry is already eligible.

For monotonicity, $d_r \leq \delta$ implies $d_r \leq \delta'$, so $\mathcal{T}_t(\delta) \subseteq \mathcal{T}_t(\delta')$. The smallest preference index over a larger set cannot increase, proving Equation (S26). At $\delta = 0$, only score maximizers are eligible and the earliest of them is selected. Once $\delta \geq d_{\sigma(1)}$, the first configured route is eligible and remains selected for all larger tolerances. Between these endpoints, decisions change only at score gaps satisfying Equation (S25). □

These conclusions explain how a positive tolerance can change the route even though the score itself is unchanged. The configured preference rank improves monotonically as the eligible set expands, with route switches occurring at the score-gap thresholds. The quality bound in Equation (S24) supplies the link to the calibration and trajectory-regret bounds above.

Perturbed scores. Suppose the measured score $\widetilde{S}_t$ differs from a nominal score S_t by at most e_t uniformly over routes. A route $\widetilde{r}_t$ in the measured tolerance set satisfies

$$\begin{aligned} \max_r S_t(r) - S_t(\widetilde{r}_t) &\leq \max_r \widetilde{S}_t(r) + e_t - \widetilde{S}_t(\widetilde{r}_t) + e_t \\ &\leq \delta_t + 2e_t. \end{aligned} \tag{S27}$$

Here e_t controls estimation noise in a specified score; the calibration error ϵ_t measures the discrepancy between that score and the population risk certificate.

The same noise model yields an exact decision-stability condition. Suppress t, keep the configured order and tolerance fixed, and let $k = k(\delta)$ be the nominal selected rank. Write $\widetilde{M} = \max_r \widetilde{S}(r)$ and $\widetilde{d}_r = \widetilde{M} - \widetilde{S}(r)$. For each r, $S(r) - e \leq \widetilde{S}(r) \leq S(r) + e$; maximizing all three expressions gives $M - e \leq \widetilde{M} \leq M + e$. Consequently,

$$|\widetilde{d}_r - d_r| \leq |\widetilde{M} - M| + |\widetilde{S}(r) - S(r)| \leq 2e. \tag{S28}$$

Suppose that the nominal selected route has $d_{\sigma(k)} \leq \delta - 2e$ and every earlier entry has $d_{\sigma(j)} > \delta + 2e$. The bound then gives $\widetilde{d}_{\sigma(k)} \leq \delta$ and $\widetilde{d}_{\sigma(j)} > \delta$ for every $j < k$. By the two directions of

Equation (S25), the perturbed selector returns the same route. Later candidates may enter or leave the tolerance set without affecting this conclusion. This sufficient condition identifies the relevant margins on both sides of the eligibility threshold; Equation (S27) continues to bound score loss even when these margins are too small to preserve the exact decision.

An explicit finite-sample specialization applies to an additive diagnostic on an independent evaluation batch. Let $\mathcal{F}_t$ contain the history and constructed candidates before that batch is drawn, and let $\widetilde{S}_t(r) = n_t^{-1} \sum_i Z_{tri}$, where the $Z_{tri} \in [a, a+B]$ are independent across i conditional on $\mathcal{F}_t$. Set $S_t(r) = \mathbb{E}[\widetilde{S}_t(r) \mid \mathcal{F}_t]$. For $e > 0$, conditional Hoeffding gives $\Pr(|\widetilde{S}_t(r) - S_t(r)| > e \mid \mathcal{F}_t) \le 2\exp(-2n_t e^2/B^2)$. Taking conditional expectations and then summing the failure probabilities over routes and stages gives $\Pr(\exists (t,r) : |\widetilde{S}_t(r) - S_t(r)| > e_t) \le \sum_t 2K \exp(-2n_t e_t^2/B^2)$. Thus, with probability at least $1-\zeta$,

$$|\widetilde{S}_t(r) - S_t(r)| \le B\sqrt{\frac{\log(2KT/\zeta)}{2n_t}} \quad \text{for all } (t,r). \tag{S29}$$

Choose e_t as the right-hand side, so each summand is ζ/T. The union bound permits dependence between routes and across stages. Clipped cross-entropy with fixed per-example targets is one bounded additive diagnostic satisfying the stated assumptions. For a route score with data-dependent masks, normalization, and centroid geometry, e_t denotes a uniform perturbation bound for that complete statistic. Substituting the corresponding e_t into Equation (S27) gives its tolerance guarantee.

Entropy characterization of the tolerance set. Let $U_t(r) = -\widehat{L}^r_{\text{bridge}} - \widehat{L}^r_{\text{smooth}}$ and $I_t(r) = \widehat{L}^r_{\text{con}}$. For an agreement-loss budget b, the constrained problem is $\max_{r: I_t(r) \le b} U_t(r)$. Its maximization Lagrangian is $U_t(r) - \tau_t(I_t(r) - b)$ with $\tau_t \ge 0$. For fixed τ_t, the constant $\tau_t b$ does not affect ranking, leaving Equation (10). If a penalized maximizer $\bar{r}$ is feasible and satisfies $\tau_t(I_t(\bar{r}) - b) = 0$, then for any feasible r, $U_t(r) \le U_t(r) - \tau_t(I_t(r) - b) \le U_t(\bar{r}) - \tau_t(I_t(\bar{r}) - b) = U_t(\bar{r})$. Feasibility and complementary slackness thus establish optimality for the constrained problem whenever such a multiplier exists. The penalty score gives a direct route-ranking criterion for any configured τ_t.

For any auxiliary temperature $\vartheta > 0$, consider a probability distribution μ over the finite route library and the strictly concave problem

$$\max_{\mu \in \Delta^{K-1}} \mathcal{F}_\vartheta(\mu), \qquad \mathcal{F}_\vartheta(\mu) = \sum_r \mu_r S_t(r) - \vartheta \sum_r \mu_r \log \mu_r. \tag{S30}$$

Continuity with $0 \log 0 = 0$ and compactness give existence. If a maximizer had $\mu_j = 0$, moving mass ε from a positive coordinate to j changes the utility by $O(\varepsilon)$ but increases entropy by $-\varepsilon \log \varepsilon + O(\varepsilon)$, a strictly positive net gain for sufficiently small ε. Thus all coordinates are positive. Introducing a multiplier ν for $\sum_r \mu_r = 1$ gives the stationarity chain

$$\begin{aligned} 0 &= S_t(r) - \vartheta(1 + \log \mu_r) + \nu, \\ \mu_r &= \exp((\nu - \vartheta)/\vartheta) \exp(S_t(r)/\vartheta), \\ \mu_r^\vartheta &= \frac{\exp(S_t(r)/\vartheta)}{Z_\vartheta}, \qquad Z_\vartheta = \sum_j \exp(S_t(j)/\vartheta). \end{aligned} \tag{S31}$$

Normalization eliminates the multiplier. Substituting $\log \mu_r^\vartheta = S_t(r)/\vartheta - \log Z_\vartheta$ gives the full identity

$$\begin{aligned} \vartheta \, \mathrm{KL}(\mu \| \mu^\vartheta) &= \vartheta \sum_r \mu_r \log \mu_r - \sum_r \mu_r S_t(r) + \vartheta \log Z_\vartheta \\ &= \vartheta \log Z_\vartheta - \mathcal{F}_\vartheta(\mu). \end{aligned} \tag{S32}$$

Nonnegativity of KL and its equality condition prove global optimality and uniqueness, including comparison with distributions on the simplex boundary.

Let $r^\dagger$ maximize S_t. Dividing two Gibbs probabilities cancels the partition function and yields

$$\begin{aligned} \frac{\mu_r^\vartheta}{\mu_{r^\dagger}^\vartheta} &= \exp\big((S_t(r) - S_t(r^\dagger))/\vartheta\big), \\ \mu_r^\vartheta \ge e^{-\delta_t/\vartheta} \mu_{r^\dagger}^\vartheta \quad &\iff \quad S_t(r) \ge S_t(r^\dagger) - \delta_t \quad \iff \quad r \in \mathcal{T}_t(\delta_t). \end{aligned} \tag{S33}$$

The equivalence holds at every positive temperature. It explains the tolerance set as a relative-probability level set, while the actual decision remains the deterministic configured preference rule in Equation (11). The temperature cancels from the eligibility condition, leaving a direct computation from score differences and δ_t.

S3.4 Prototype bridging and confidence filtering

Fix one input x and suppress its argument. Let $q_j \in \Delta^{C-1}$ and $w_j \geq 0$ with $\sum_j w_j = 1$. Write $\bar{q} = \sum_j w_j q_j$ and $B(q) = \frac{1}{2}\sum_j w_j \|q - q_j\|_2^2$. Expanding around $\bar{q}$ gives

$$\begin{aligned} 2B(q) &= \sum_j w_j \|q - \bar{q} + \bar{q} - q_j\|_2^2 \\ &= \|q - \bar{q}\|_2^2 + 2\left\langle q - \bar{q}, \sum_j w_j(\bar{q} - q_j)\right\rangle + \sum_j w_j \|\bar{q} - q_j\|_2^2 \\ &= \|q - \bar{q}\|_2^2 + 2B(\bar{q}). \end{aligned} \tag{S34}$$

The cross term vanishes because $\sum_j w_j q_j = \bar{q}$. The remaining squared norm is nonnegative and zero exactly at $q = \bar{q}$. Since a convex combination of simplex points lies in the simplex, the unconstrained minimizer is feasible and unique. Substituting the three streams proves Equation (6).

The distribution-level functional is $\mathfrak{B}_r[q] = \mathbb{E}_{P_{t+1}} B_r(q(x))$. For a square-integrable tangent perturbation σ with $\mathbf{1}^\top \sigma(x) = 0$, its first variation is

$$\begin{aligned} \delta\mathfrak{B}_r[q;\sigma] &= \frac{d}{d\varepsilon}\mathbb{E}\frac{1}{2}\sum_j w_{r,j}\|q + \varepsilon\sigma - q_j\|_2^2\Big|_{\varepsilon=0} \\ &= \mathbb{E}\left\langle \sum_j w_{r,j}(q - q_j), \sigma\right\rangle = \mathbb{E}\langle q - \widetilde{q}_r, \sigma\rangle. \end{aligned} \tag{S35}$$

The exchange of derivative and expectation follows directly by expanding the quadratic polynomial in ε; its coefficients are integrable because simplex-valued predictions are bounded and σ is square-integrable. The second variation is $\mathbb{E}\|\sigma\|_2^2$, so the minimizer is unique up to P_{t+1}-null sets. At a simplex boundary, feasible one-sided perturbations define the variation. The completed-square identity in Equation (S34) establishes global optimality over the entire probability simplex, including its boundary.

The minimum value can also be expressed through pairwise disagreement:

$$B(\bar{q}) = \frac{1}{4}\sum_{j,k} w_j w_k \|q_j - q_k\|_2^2. \tag{S36}$$

Indeed, expanding the right-hand side yields $\frac{1}{2}\sum_j w_j\|q_j\|^2 - \frac{1}{2}\|\bar{q}\|^2$, which is the expansion of $B(\bar{q})$. The identity explains why different mixtures respond differently to disagreement among source, prototype, and current prediction streams.

If the streams are perturbed to q'_j, convexity of the norm gives

$$\|\bar{q} - \bar{q}'\|_1 = \left\|\sum_j w_j(q_j - q'_j)\right\|_1 \leq \sum_j w_j\|q_j - q'_j\|_1. \tag{S37}$$

If only the weights change to w', the fact that $\|q_j\|_1 = 1$ instead yields $\|\bar{q}(w) - \bar{q}(w')\|_1 \leq \|w - w'\|_1$. Together these inequalities make route weights interpretable: changing a stream or a mixture changes the bridge in proportion to the corresponding perturbation, before pseudo-label thresholding. Hard masks can be discontinuous at their thresholds, which is why the algorithm defines mask decisions separately from the barycenter calculation.

Loss sensitivity to the bridge weights. For fixed streams and a fixed target vector $y \in \Delta^{C-1}$, eliminate $\gamma = 1-\alpha-\beta$ and write $\widetilde{q} = q_t + \alpha(q_s - q_t) + \beta(q_p - q_t)$. Define $\ell_y(\alpha,\beta) = \mathrm{CE}(y,\widetilde{q})$ on $\alpha,\beta \geq 0$, $\alpha+\beta \leq 1$, with $\widetilde{q}_c > 0$ whenever $y_c > 0$. All sums in this weight-sensitivity calculation omit coordinates with $y_c = 0$. Convexity of $-\log$ gives the componentwise Jensen bound

$$\begin{aligned}\ell_y(\alpha,\beta) &= -\sum_c y_c \log\left(\alpha q_{s,c} + \beta q_{p,c} + \gamma q_{t,c}\right) \\ &\leq \alpha\,\mathrm{CE}(y,q_s) + \beta\,\mathrm{CE}(y,q_p) + \gamma\,\mathrm{CE}(y,q_t). \end{aligned} \tag{S38}$$

Terms with zero weights are omitted. For finite stream losses, equality requires the positively weighted streams to assign the same probability to every class with $y_c > 0$.

Set $d_{s,c} = q_{s,c} - q_{t,c}$ and $d_{p,c} = q_{p,c} - q_{t,c}$. Since $\partial_\alpha \widetilde{q}_c = d_{s,c}$ and $\partial_\beta \widetilde{q}_c = d_{p,c}$, differentiating $-y_c \log \widetilde{q}_c$ once gives $-y_c d_{s,c}/\widetilde{q}_c$; differentiating that term in β gives $y_c d_{s,c} d_{p,c}/\widetilde{q}_c^2$. Summing yields

$$\frac{\partial \ell_y}{\partial \alpha} = -\sum_c \frac{y_c d_{s,c}}{\widetilde{q}_c}, \qquad \frac{\partial \ell_y}{\partial \beta} = -\sum_c \frac{y_c d_{p,c}}{\widetilde{q}_c}, \qquad \nabla^2 \ell_y = \sum_c \frac{y_c}{\widetilde{q}_c^2} \begin{pmatrix} d_{s,c}^2 & d_{s,c} d_{p,c} \\ d_{s,c} d_{p,c} & d_{p,c}^2 \end{pmatrix}. \tag{S39}$$

For any direction $v = (v_1, v_2)$, multiplying on both sides yields $v^\top \nabla^2 \ell_y v = \sum_c y_c (v_1 d_{s,c} + v_2 d_{p,c})^2 / \widetilde{q}_c^2 \geq 0$. Hence the bridge cross-entropy is jointly convex in its two free weights on positive support. The mixed partial derivatives complete the joint curvature calculation for the bridge weights.

For a one-hot target $y = e_k$, the first derivative is negative exactly when the source stream assigns more probability to class k than the current stream does. The prototype derivative has the analogous interpretation. Thus each route weight changes supervision in a direction determined by disagreement between semantic streams. These derivatives describe fixed targets and streams; recomputing pseudo-labels or masks changes the objective piecewise, and updating the network changes the streams. The result concerns local probability mixing, while the trajectory comparison remains the policy program in Equation (2).

Coverage-constrained confidence filtering. Fix a route and its scalar confidence function $g(x)$ on the current input marginal P. For a prescribed coverage $\chi \in [0,1]$, the relaxed mask solves

$$\max_{0 \leq m(x) \leq 1} \mathbb{E}_P[m(x) g(x)] \quad \text{subject to} \quad \mathbb{E}_P[m(x)] = \chi. \tag{S40}$$

Its Lagrangian is $\Lambda(m,\nu) = \mathbb{E}_P[m(x)(g(x)-\nu)] + \nu\chi$. The variation in a feasible direction σ is $\mathbb{E}_P[(g-\nu)\sigma]$. Pointwise maximization of this linear functional therefore gives

$$m^*(x) = \begin{cases} 1, & g(x) > \nu, \\ 0, & g(x) < \nu, \\ a(x) \in [0,1], & g(x) = \nu, \end{cases} \qquad \mathbb{E}_P[m^*] = \chi. \tag{S41}$$

Here $g \in [0,1]$. For $0 < \chi < 1$, choose a quantile ν such that $\Pr(g > \nu) \leq \chi \leq \Pr(g \geq \nu)$. If $\Pr(g = \nu) > 0$, set $a(x) = (\chi - \Pr(g > \nu))/\Pr(g = \nu)$ on that atom; otherwise no boundary allocation is needed. This constructs a feasible optimizer even for discrete confidences. At $\chi = 0$ and $\chi = 1$, the only feasible masks are respectively zero and one almost surely.

An exchange argument establishes optimality for both discrete and continuous confidence distributions. For any feasible m, both masks have the same coverage, and

$$\begin{aligned}\mathbb{E}_P[(m^* - m) g] &= \mathbb{E}_P[(m^* - m)(g - \nu)] + \nu \mathbb{E}_P[m^* - m] \\ &= \mathbb{E}_P[(m^* - m)(g - \nu)] \geq 0. \end{aligned} \tag{S42}$$

Above the threshold both factors are nonnegative, below it both are nonpositive, and at it their product vanishes. In an empirical class-conditional sample, the same argument retains the highest-scoring entries within each predicted class. If the desired mass is not an integer multiple of the sample weight, the relaxed optimum uses a fractional boundary entry; the hard-mask convention in Equation (S2) rounds the retained count and resolves ties by index.

Applying the program separately to $P(\cdot \mid \widetilde{y}_r = c)$ explains the class-wise quantile construction. It prevents a fixed global confidence threshold from determining every class's retained count solely through differences in confidence scale. Subsequent agreement filtering can reduce these counts. The quantile rule maximizes retained confidence at the prescribed coverage using the current unlabeled predictions. The retained coverage enters the risk certificate through ρ_t, so the derivation also identifies the tradeoff between more selective supervision and the uncovered-sample term.

S3.5 SUPERVISED SMOOTHING AND SEMANTIC STABILITY

All bridge labels and masks are held fixed during a candidate optimization block. This convention makes the objective in Equation (9) a differentiable loss of the student parameters wherever the student itself is differentiable. Gradients flow through the student predictions with the generated targets held constant.

For logits $z \in \mathbb{R}^C$, probabilities $p = \mathrm{softmax}(z)$, and a fixed target probability vector q, the cross-entropy is $\mathrm{CE}(q,p) = -\sum_c q_c \log p_c$. Since $\log p_c = z_c - \log \sum_j e^{z_j}$,

$$\frac{\partial\, \mathrm{CE}(q,p)}{\partial z_j} = -q_j + \left(\sum_c q_c\right) \frac{e^{z_j}}{\sum_k e^{z_k}} = p_j - q_j. \tag{S43}$$

For a true source label, $q = e_y$ is externally supplied. The source gradient therefore has a fixed class reference even if target predictions are confidently wrong. For target pseudo-labels, the same formula uses a detached $e_{\tilde{y}_r}$ and can reinforce an incorrect class. Source supervision therefore supplies a persistent semantic anchor, while detached target labels guide adaptation to the current domain.

For the alignment term, $\mathrm{KL}(q\|p) = \mathrm{CE}(q,p) - H(q)$. The entropy $H(q)$ is constant during the student update, so the alignment gradient is also $p-q$, while its minimizer in unconstrained probability space is $p = q$. The two terms provide complementary supervision: pseudo-label fitting emphasizes the discrete class, and alignment retains uncertainty across classes.

Now consider V view predictions p_v and their average $\bar{p}$. The generalized Jensen–Shannon identity is

$$\begin{aligned}\frac{1}{V}\sum_v \mathrm{KL}(p_v\|\bar{p}) &= \frac{1}{V}\sum_{v,c} p_{v,c}\log p_{v,c} - \sum_c \bar{p}_c \log \bar{p}_c \\ &= H(\bar{p}) - \frac{1}{V}\sum_v H(p_v) \geq 0.\end{aligned} \tag{S44}$$

Nonnegativity follows from KL divergence and equivalently from concavity of entropy, and equality holds exactly when all views have the same predictive distribution. Because $H(\bar{p}) \leq \log C$, each per-sample agreement loss lies in $[0, \log C]$. Averaging over retained samples preserves this range, and the empty-support convention sets the loss to zero. Pinsker's inequality further implies

$$\frac{1}{V}\sum_v \mathrm{TV}(p_v,\bar{p}) \leq \frac{1}{V}\sum_v \sqrt{\mathrm{KL}(p_v\|\bar{p})/2} \leq \sqrt{\frac{1}{2V}\sum_v \mathrm{KL}(p_v\|\bar{p})}. \tag{S45}$$

The last step is Jensen's inequality for the square root. Agreement therefore controls average prediction displacement across the specified views, quantifying the consistency encouraged by the multi-view loss.

For a fixed stage objective $F(\theta)$ with L-Lipschitz gradient, gradient descent $\theta^+ = \theta - \eta\nabla F(\theta)$ satisfies the standard descent chain

$$\begin{aligned}F(\theta^+) &\leq F(\theta) + \langle \nabla F(\theta), \theta^+ - \theta\rangle + \frac{L}{2}\|\theta^+ - \theta\|_2^2 \\ &= F(\theta) - \eta\left(1 - \frac{L\eta}{2}\right)\|\nabla F(\theta)\|_2^2.\end{aligned} \tag{S46}$$

Thus $0 < \eta < 2/L$ ensures descent for a deterministic gradient step with fixed masks and an L-smooth objective. This local student optimization complements probability-space barycenter optimality and the route-policy comparison.

Source anchoring and accumulated perturbations. A conditional local model makes the anchoring effect precise. Let F_s be a differentiable source objective with reference point θ_s, where $\nabla F_s(\theta_s) = 0$. Assume in the region visited by the update that $\langle \theta - \theta_s, \nabla F_s(\theta)\rangle \geq \mu\|\theta - \theta_s\|_2^2$ for some $\mu > 0$. This is a strong-monotonicity condition, satisfied for example by a locally strongly convex source objective. Let the target-side gradient contribution be $g_u(\theta, u)$ with norm bounded by $b(u)$, and consider the local gradient flow

$$\dot{\theta}(u) = -\lambda_s \nabla F_s(\theta(u)) - g_u(\theta(u), u), \qquad \lambda_s > 0. \tag{S47}$$

Here u is optimization time within a stage. Differentiating the squared distance and applying the two assumptions gives

$$\frac{1}{2}\frac{d}{du}\|\theta(u)-\theta_s\|_2^2 = -\lambda_s\langle\theta-\theta_s,\nabla F_s(\theta)\rangle - \langle\theta-\theta_s,g_u\rangle \\ \le -\lambda_s\mu\|\theta-\theta_s\|_2^2 + b(u)\|\theta-\theta_s\|_2. \tag{S48}$$

Assume the flow exists and remains in the stated region, with locally integrable b. For $d(u) = \|\theta(u)-\theta_s\|_2$, division where $d>0$ gives $\dot d \le -\lambda_s\mu d + b(u)$; the upper directional derivative gives the same inequality at zero. With $a=\lambda_s\mu$, the product rule gives $\frac{d}{du}(e^{au}d(u)) \le e^{au}b(u)$ almost everywhere. Integrating from 0 to u and multiplying by e^{-au} yields

$$d(u) \le e^{-\lambda_s\mu u}d(0) + \int_0^u e^{-\lambda_s\mu(u-v)}b(v)\,dv. \tag{S49}$$

If $b(v)\le \bar b$, the integral is at most $\bar b(1-e^{-\lambda_s\mu u})/(\lambda_s\mu)$. The source term contracts inherited displacement in this local model, while target updates contribute a bounded forced response. Iterating stage bounds $d_{t+1}\le a_t d_t + b'_t$ gives

$$d_T \le \left(\prod_{t=0}^{T-1} a_t\right) d_0 + \sum_{j=0}^{T-1} b'_j \prod_{k=j+1}^{T-1} a_k. \tag{S50}$$

Here $a_t = e^{-\lambda_s\mu u_t}$ for a stage duration u_t under a common local reference, and b'_t is its integrated target perturbation. This conditional calculation explains the damping role of source anchoring. The semantic label reference follows directly from Equation (S43); contraction additionally requires the stated local geometry. Equation (S57) treats general complete update operators, including settings outside this gradient-flow model.

S3.6 Target-risk guarantees and path perturbations

We use $\ell(p,y)=1-p_y\in[0,1]$ and natural logarithms. All population expectations are conditional on the constructed model, bridge, and measurable mask rule, with a fresh input from P_{t+1}. For a quantile policy, the construction sample fixes its cutoffs and tie rule before this evaluation. This defines population coverage separately from the empirical retained count. For probability vectors p,q, $\sum_c(p_c-q_c)=0$, so each coordinate difference is bounded by the sum of positive differences, which equals $\mathrm{TV}(p,q)=\frac{1}{2}\sum_c|p_c-q_c|$. For any pre-update h and post-update h', adding and subtracting the next-domain risk of h gives

$$R_{t+1}(h')-R_t(h) = R_{t+1}(h')-R_{t+1}(h)+R_{t+1}(h)-R_t(h) \\ \le \mathbb{E}_{(x,y)\sim\mathcal{D}_{t+1}}|h'(x)_y-h(x)_y| + D_t(h) \\ \le \mathbb{E}_{x\sim P_{t+1}}\mathrm{TV}(h'(x),h(x)) + D_t(h). \tag{S51}$$

The inequality holds conditional on the realized model and its route history. The domain-change term captures shifts in the joint input-label distribution, including changes in the labeling rule.

Proof of Theorem 3.2. Let $q=\widetilde q_{r_t^*}$ and $m=m_{r_t^*}$ for the selected stage. The triangle inequality and the retained/unretained partition give

$$\mathbb{E}\,\mathrm{TV}(h_{t+1},h_t) \le \mathbb{E}\,\mathrm{TV}(q,h_t)+\mathbb{E}\,\mathrm{TV}(h_{t+1},q) \\ = b_t + \mathbb{E}[m\,\mathrm{TV}(h_{t+1},q)] + \mathbb{E}[(1-m)\,\mathrm{TV}(h_{t+1},q)] \\ \le b_t + \mathbb{E}[m\sqrt{\mathrm{KL}(q\|h_{t+1})/2}] + 1-\rho_t \\ \le b_t + \sqrt{\mathbb{E}[m]\,\mathbb{E}[m\,\mathrm{KL}(q\|h_{t+1})]/2} + 1-\rho_t \\ = b_t + \rho_t\sqrt{a_t/2} + 1-\rho_t. \tag{S52}$$

The third line uses Pinsker on retained samples and $\mathrm{TV}\le 1$ on the complement. The fourth uses Cauchy–Schwarz on $\sqrt{m}$ and $\sqrt{m\,\mathrm{KL}/2}$. For $\rho_t>0$, $\mathbb{E}[m\,\mathrm{KL}]=\rho_t a_t$, giving the last line. For $\rho_t=0$, the retained contribution is zero; for $\rho_t=1$, the complement is zero. Both endpoints follow by evaluating the retained and complementary contributions directly. Products involving $m=0$

are defined as zero before evaluating KL. Substituting into Equation (S51) defines the nonnegative certificate

$$c_t(s_t, r_t^*) = D_t(h_t) + b_t + \rho_t\sqrt{a_t/2} + 1 - \rho_t. \tag{S53}$$

The same construction defines $c_t(s, r)$ for every candidate: use its bridge, mask, and complete post-update classifier $h_{\mathcal{G}_t^r(s)}$. The finite-cost policy analysis assumes finite retained KL moments at admissible states; a uniform positive lower bound on student probabilities suffices. Summing gives $\sum_{t=0}^{T-1}[R_{t+1}(h_{t+1}) - R_t(h_t)] = R_T(h_T) - R_0(h_0) \le \sum_t c_t$, which is Equation (14). □

The certificate shows what each type of control can affect. Bridge construction controls b_t; bridge alignment controls a_t; retaining adequate support controls $1-\rho_t$; and the domain sequence determines D_t. The topology and agreement losses influence these quantities through the learned predictor and retention masks. With target top-1 prediction $\widehat{y} = \arg\max_c h_c$, one has $\mathbf{1}\{\widehat{y} \neq y\} \le 2(1 - h_y)$, because a wrongly predicted true class has probability at most $1/2$. This gives a corresponding upper bound on top-1 error. The experiments evaluate top-1 accuracy directly on the target examples.

Policy optimization under the certificate. For every admissible policy π, the certificate gives $R_T(h_T^\pi) \le J(\pi)$. With finite routes and horizon, let π^* minimize J conditional on the fixed random streams. Combining Theorem 3.2 with policy inclusion yields $R_T(h_T^{\pi^*}) \le J^*_{\text{relax}} \le J^*_{\text{fixed}}$. For the selected policy, Theorem 3.1 bounds $J(\widehat{\pi})$ by the adaptive optimum plus the accumulated tolerance, calibration, and continuation terms. These relations connect route relaxation and local selection to the final pathwise risk bound.

Class topology and geometric control. The loss in Equation (8) penalizes changes in inter-class centroid geometry. If $m = |\mathcal{P}_r| > 0$ and $L^r_{\text{topo}} \le \varepsilon^2$, then

$$|G^r_{cd} - G^s_{cd}| \le \sqrt{m}\,\varepsilon \quad \text{for every } (c, d) \in \mathcal{P}_r, \tag{S54}$$

because each squared difference is one nonnegative summand of a sum bounded by $m\varepsilon^2$. If the reference squared separations satisfy $G^s_{cd} \ge g > 0$, the retained centroids remain separated by at least $\sqrt{\max(0, g - \sqrt{m}\varepsilon)}$. This is an explicit separation guarantee for the centroid geometry.

A conditional nearest-centroid consequence makes the role of sample spread precise. Suppose each example of class c lies within radius a of its correctly indexed centroid, and all inter-class centroid distances exceed $2a$. For any other class d,

$$\|z - p_d\|_2 \ge \|p_c - p_d\|_2 - \|z - p_c\|_2 > 2a - a = a \ge \|z - p_c\|_2. \tag{S55}$$

The nearest centroid is then the correct one. Combining this condition with Equation (S54) explains how preserved class separation can support a bridge. Source labels and source-anchored bridge probabilities associate centroids with class identities. Together with the assumed sample-spread bound, these semantic anchors connect preserved geometric separation to the nearest-centroid decision.

Route and domain-order perturbations. Equip states and routes with distances d_s and d_r to quantify how route perturbations propagate. Suppose

$$\begin{aligned} d_s(\mathcal{G}_t^r(s), \mathcal{G}_t^{r'}(s')) &\le L_t d_s(s, s') + B_t d_r(r, r'), \\ |c_t(s, r) - c_t(s', r')| &\le A_t d_s(s, s') + C_t d_r(r, r'). \end{aligned} \tag{S56}$$

The assumed constants bound the sensitivity of the complete update and stage cost with respect to the retained state and route. For two selected route sequences with $e_t = d_s(s_t, s'_t)$ and $u_t = d_r(r_t, r'_t)$, Equation (S56) gives $e_{t+1} \le L_t e_t + B_t u_t$. With a common initial state, repeated substitution yields

$$\begin{aligned} e_1 &\le B_0 u_0, \\ e_2 &\le L_1 B_0 u_0 + B_1 u_1, \\ e_t &\le \sum_{j=0}^{t-1} B_j u_j \prod_{k=j+1}^{t-1} L_k. \end{aligned} \tag{S57}$$

The empty product is one. The formula follows by induction: multiply the expression at t by L_t and add $B_t u_t$ to obtain the expression at $t+1$. It displays the greater opportunity for an early perturbation to propagate through later transitions.

Summing the cost comparison and exchanging the finite sums gives

$$|J(\mathbf{r}) - J(\mathbf{r}')| \le \sum_{t=0}^{T-1} (A_t e_t + C_t u_t)$$
$$\le \sum_{j=0}^{T-1} \left[C_j + B_j \sum_{t=j+1}^{T-1} A_t \prod_{k=j+1}^{t-1} L_k \right] u_j. \tag{S58}$$

The bracketed coefficient contains an immediate cost and the accumulated influence on future costs. Its accumulated terms quantify how the position of a route decision and the sensitivity of subsequent updates jointly determine its effect.

An analogous statement handles an imperfect supplied domain order. Let $\widetilde{\mathcal{G}}_t$ and $\widetilde{c}_t$ denote the operators and costs under a perturbed order. Suppose their stagewise differences at the same state and route are bounded by ξ_t and χ_t, respectively. Allow the policies to choose different routes under the two orders. Then

$$e_{t+1} \le L_t e_t + B_t u_t + \xi_t,$$
$$|J - \widetilde{J}| \le \sum_t (A_t e_t + C_t u_t + \chi_t),$$
$$e_t \le \sum_{j<t} (B_j u_j + \xi_j) \prod_{k=j+1}^{t-1} L_k. \tag{S59}$$

The proof adds and subtracts $\mathcal{G}_t^{r'_t}(s'_t)$ in the state distance and $c_t(s'_t, r'_t)$ in the cost difference, then applies the triangle inequality and the preceding induction. Thus policy inclusion remains valid for any fixed supplied order, while the attainable risk certificate and regret depend on that order through the state transitions and domain-change terms. Larger ξ_t, χ_t, or D_t quantify the effect of a misleading intermediate domain, with subsequent update sensitivities determining its propagation.

S4 Detailed Experimental Evidence

S4.1 Benchmark Protocols and Metrics

The primary results in Tables 1, 2, and 3 evaluate the ordered-benchmark settings described in Section 4. The additional experiments vary the number of domains and, for Portraits, image resolution. Table S4 summarizes these configurations. Here the number of domains includes both endpoints, so m domains correspond to $m - 1$ transitions. The six-domain corruption path in Table S4 consists of clean source data followed by five severity levels.

Table S4: Supplemental benchmark configurations.

Benchmark	Classes	Domain settings	Ordering variable
Rotated MNIST	10	2/4/6	Rotation, maximum 45°
Color-Shift MNIST	10	2/4/6	Monotone appearance change
Portraits, 32×32	2	2/4/6	Chronology
Portraits, 64×64	2	6/9	Chronology
Portraits, 96×96	2	6/9	Chronology
Covertype	2	2/4/6	Distance to water
CIFAR-10-C	10	6	Severity 0–5, per corruption
CIFAR-100-C	100	6	Severity 0–5, per corruption
Office-31	31	3	Given visual-domain path

Portraits is a binary gender-classification task on chronologically ordered portrait collections. Its temporal splits and resolution variants preserve the same task while varying the visual evidence available at each transition. Covertype uses a binary label task with intermediate domains indexed by distance to water. These definitions distinguish the prediction labels from the variable that orders the domains.

For a target evaluation set with N labels available only for scoring, accuracy is $100N^{-1}\sum_i \mathbf{1}\{\arg\max_c h_T(x_i)_c = y_i\}$. Error is 100 minus accuracy. A corruption mean is the equally weighted average across the 15 corruption types. A corruption-family mean averages only the types in that family: three noise, four blur, four weather, and four digital corruptions. A mean across path settings assigns equal weight to the evaluated adaptation configurations.

S4.2 Severity-5 comparison protocols and aggregation

Table S5 summarizes the mean accuracies and adaptation information for the complete CIFAR-10-C and CIFAR-100-C comparisons in Tables 2 and 3. All 300 values use the same accuracy definition, and each method's mean weights the 15 corruption types equally. The method labels identify source-only evaluation, batch-normalization adaptation, continual test-time adaptation, and ordered adaptation. The severity-path column distinguishes access to intermediate domains from adaptation on the current test distribution. Tent, AdaCon., and GTTA in the main tables denote Tent-cont., AdaContrast, and GTTA-MIX, respectively.

Table S5: Mean severity-5 accuracy (%, ↑) and access to intermediate severities.

Method	Severity path	CIFAR-10-C	CIFAR-100-C
Source only	No	56.5	53.6
BN-1	No	79.6	64.6
Tent-cont. (Wang et al., 2021)	No	79.3	39.1
AdaContrast (Chen et al., 2022b)	No	81.5	66.6
CoTTA (Wang et al., 2022b)	No	83.8	67.5
GTTA-MIX (Marsden et al., 2024)	No	84.4	71.1
GST (Kumar et al., 2020)	Yes	72.0	66.0
GOAT (He et al., 2024)	Yes	56.6	53.9
STDW (Wang et al., 2026)	Yes	85.2	74.2
PrismBridge	Yes	**93.3**	**90.2**

S4.3 Isolating route size, tolerance, and path position

The library-size, tolerance, early-route, path-density, and hyperparameter studies report means over five random seeds. The path-density and hyperparameter figures additionally show the corresponding 95% confidence bands. Each control changes the named factor while retaining the other components of the complete framework.

Tables S6 and S7 expand the mechanism diagnostics in Figure 3. Increasing the library from one to four routes improves Rotated MNIST by 2.55 points, Color-Shift MNIST by 3.18 points, Portraits by 3.95 points, and Covertype by 5.71 points. Gains from three to four routes are smaller, and six routes does not improve the four-route result. The pattern supports a compact library spanning complementary semantic emphases.

Table S6: Route-library size and target accuracy (%).

Dataset	1	2	3	4	6
Rotated MNIST	93.92	95.31	96.02	96.47	96.36
Color-Shift MNIST	96.02	97.88	98.67	99.20	99.05
Portraits	89.40	91.06	92.48	93.35	93.02
Covertype	76.31	79.08	81.24	82.02	81.77

The tolerance intervention changes which score differences are treated as decisive before applying the configured route preference. The nominal scale improves over zero tolerance by 2.51 points on Portraits and 3.96 points on Covertype. Doubling it gives 92.76 and 81.13, below the nominal 93.35 and 82.02. Equation (S25) explains how enlarging the tolerance set can activate an earlier preferred route with a lower score. The measurements favor a moderate radius, balancing exclusion

by measured score against the configured preference. The global-tolerance control further supports adapting the radius to stage-dependent differences in route scores.

Table S7: Tolerance-scale sensitivity and target accuracy (%).

Dataset	0	0.5x	1x	2x	Global
Portraits	90.84	92.41	93.35	92.76	91.55
Covertype	78.06	80.91	82.02	81.13	79.82

The early-route experiment replaces selection by the default fixed route in one transition block while retaining the rest of the procedure. The seven-domain path has six transitions, divided into early, middle, and late blocks. Table 5 shows that the strongest effect occurs when early routes are fixed. On Covertype, the resulting 5.18-point drop exceeds the 2.27-point drop from fixing late routes. Selecting only early routes also exceeds selecting only late routes by 2.65 points. These interventions support the state-propagation mechanism; they perturb the route policy while keeping the supplied domain order.

S4.4 Route-policy isolation with identical stabilizers

Table S8 isolates the route decision on a six-domain Rotated-MNIST path with five transitions and a five-candidate library. The backbone, initialization, candidate routes, views, source supervision, prototype bridge, topology and consistency terms, optimization budget, and random seed are held fixed. Adaptive selection uses Equation (11); greedy selection maximizes the current score at each stage. Each fixed policy repeats its named candidate at all transitions, whereas the random policy chooses routes without score-based selection. The fixed-candidate names identify the mixture and smoothing configurations repeated along this path. Holding the remaining components fixed makes the effect of the route policy directly observable.

Table S8: Target accuracy (%) under different route policies on six-domain Rotated MNIST.

Route policy	Accuracy (%)
Adaptive stagewise	96.42
Greedy stagewise	96.48
Fixed relaxed	96.47
Fixed smooth	96.39
Random stagewise	9.74
Fixed sharpen	9.87

The adaptive policy obtains 96.42%; greedy selection, the fixed relaxed configuration, and the fixed smooth configuration obtain 96.48%, 96.47%, and 96.39%, respectively. Their full spread is 0.09 percentage points on this regular rotation path. In contrast, random selection reaches 9.74% and the fixed sharpen configuration reaches 9.87%, despite retaining all stabilizing components. The adaptive policy exceeds these two controls by 86.68 and 86.55 points. The contrast establishes sensitivity to the route policy under the same objective and training budget, while the competitive fixed configurations show that a regular path can remain compatible with one route.

The temporal intervention in Table 5 tests the complementary setting in which route suitability changes across stages. Fixing the early block lowers Portraits by 4.41 points and Covertype by 5.18 points, with smaller reductions when only middle or late blocks are fixed. Taken together, the experiments distinguish reliable individual updates from selecting an appropriate sequence of updates: all policies in Table S8 share the stabilizers, and the temporal controls preserve those components while changing where selection is allowed.

S4.5 Path density and hyperparameter sensitivity

Figure S1 evaluates the number of supplied domains, including endpoints, over the settings $\{2, 3, 4, 5, 7\}$. All four benchmark means improve as the path becomes denser. Portraits increases

from 81.00 to 93.35 and Covertype from 71.70 to 82.02. The gain from five to seven domains is smaller than the early increases, suggesting diminishing returns once local transitions become sufficiently mild. These density measurements concern the given order; the theoretical treatment of order perturbations is in Appendix S3.6.

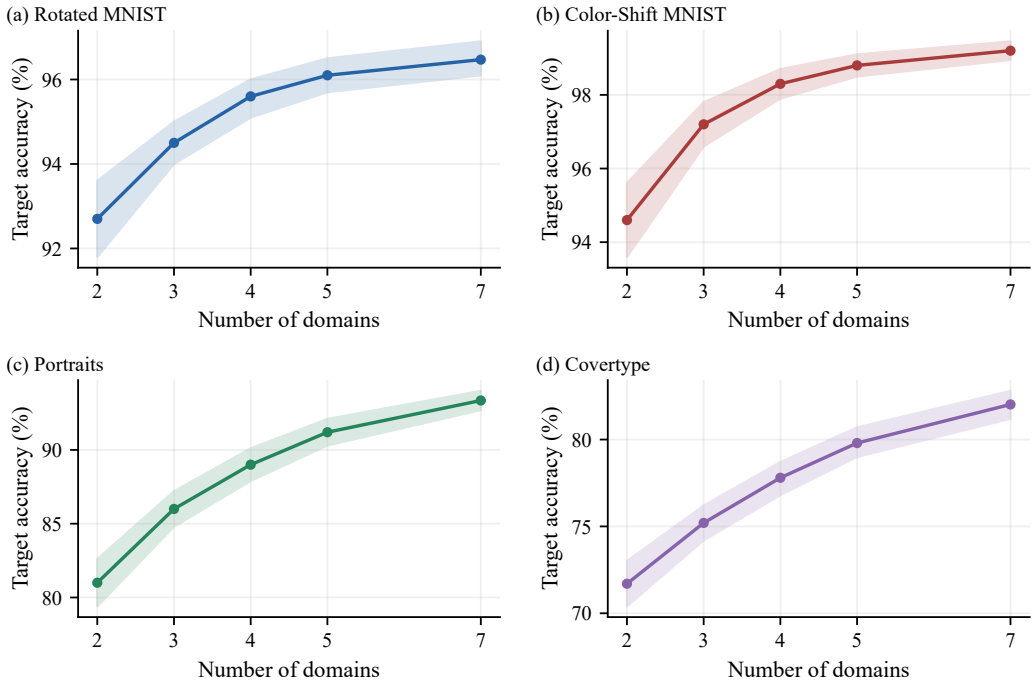


Figure S1: Path-density sensitivity on (a) Rotated MNIST, (b) Color-Shift MNIST, (c) Portraits, and (d) Covertype. Points are five-seed means; shaded regions indicate 95% confidence intervals.

Figure S2 varies one hyperparameter at a time on Portraits. The source-supervised term balances class anchoring and target responsiveness: insufficient weight weakens the anchor, whereas a large weight restricts adaptation. The topology term exhibits a related balance between preserving class geometry and permitting domain-specific movement. The stability weight influences candidate ranking, and batch size controls the number of examples supporting each diagnostic. Table S3 gives the evaluated values. In Figure S2(d), the batch-size mean increases from 90.85 at 32 examples to 93.35 at 128, then reaches 93.05 at 256.

S4.6 Effects of path length and image resolution

Table S9 reports the complete comparisons across the specified 2-, 4-, and 6-domain configurations. The controlled rotation path illustrates a large change with path length: PrismBridge reaches 51.57 with two domains and 96.40 with six. Covertype changes more modestly, from 71.54 to 73.50. The relative benefit of a denser path varies across benchmarks. In particular, SWAT achieves the highest six-domain Color-Shift MNIST value, while PrismBridge has the highest four- and six-domain Covertype values in this suite.

Table S10 extends Portraits to higher resolutions and longer paths. At 32×32, PrismBridge improves from 69.10 with two domains to 81.45 with six. At 64×64 with six domains, it reaches 88.60 versus 86.40 for STDW and 74.00 for SWAT. At 96×96 with six domains, STDW attains 91.40 versus 90.50 for PrismBridge; with nine domains the values are 90.45 and 90.95. Table S10 shows how resolution and path length jointly shape Portraits performance, with gains from additional domains varying across image resolutions.

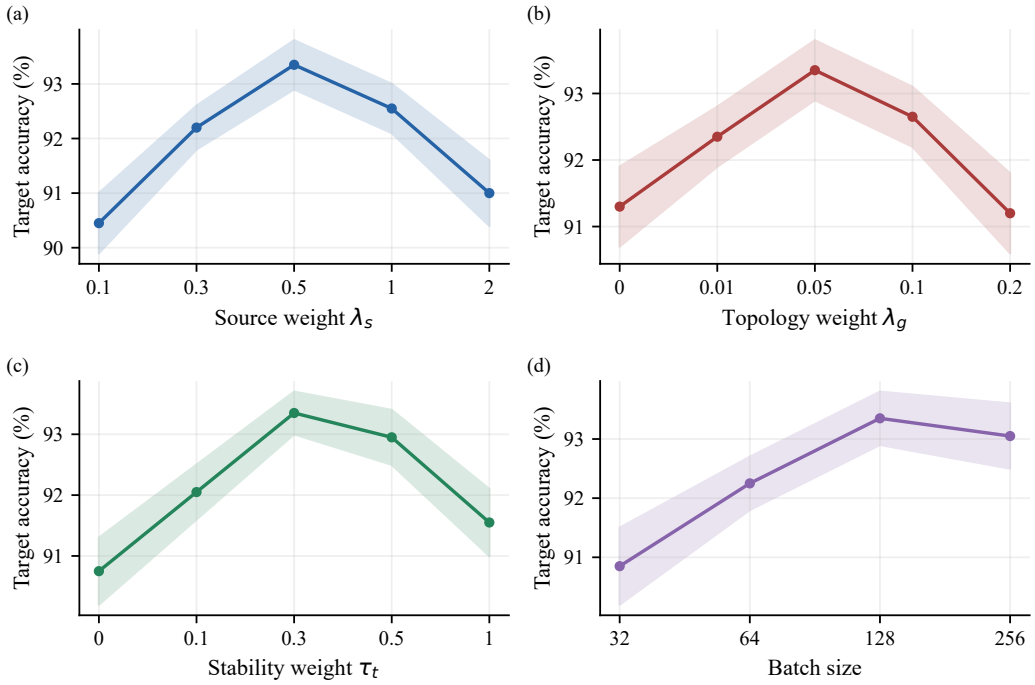


Figure S2: Portraits sensitivity to (a) source supervision, (b) topology regularization, (c) route stability weight, and (d) batch size. Points are means over five seeds; shaded regions indicate 95% confidence intervals.

Table S9: Gradual domain adaptation benchmarks. All entries are accuracies (%).

Dataset	Domains	PrismBridge	STDW	NTK	GST	SWAT
Rotated MNIST	2	51.57	44.71	**56.50**	53.83	56.02
Rotated MNIST	4	94.44	**94.84**	65.32	57.60	56.47
Rotated MNIST	6	**96.40**	96.37	77.58	61.63	56.05
Color-shift MNIST	2	99.54	**99.55**	64.38	79.76	86.60
Color-shift MNIST	4	97.78	97.79	65.16	22.61	**98.89**
Color-shift MNIST	6	97.36	97.32	66.85	17.89	**99.87**
Covertype	2	71.54	**71.80**	69.83	69.34	70.33
Covertype	4	**73.39**	72.77	69.75	69.02	70.47
Covertype	6	**73.50**	72.90	69.90	70.02	70.70

Table S10: Portraits scaling with higher image resolutions and longer domain paths. A dash denotes an unreported method–configuration combination.

Resolution	Domains	PrismBridge	STDW	SWAT
32x32	2	69.10	69.25	**69.65**
32x32	4	**72.60**	71.95	70.80
32x32	6	**81.45**	79.55	70.35
64x64	6	**88.60**	86.40	74.00
64x64	9	**87.65**	87.55	–
96x96	6	90.50	**91.40**	–
96x96	9	**90.95**	90.45	–

S4.7 MEASURED RUNTIME

Table S11 gives average end-to-end elapsed times per completed run, including route computation, for the additional experiments. PrismBridge takes 11.6 s on Portraits, 596.9 s on CIFAR-10, and 488.2 s on CIFAR-100, versus 10.2, 576.3, and 326.5 s for STDW. The ratios are 1.14, 1.04, and 1.50. On Office-31, PrismBridge takes 144.7 s versus 206.3 s for STDW and 401.6 s for SWAT. These observations show that route overhead depends on the benchmark. Section S2.5 separates shared prediction, probability mixtures, and candidate optimization, giving the dependence on route count, view count, and trial-update budget.

Table S11: Mean end-to-end runtime (seconds) for the supplemental experiments.

Benchmark	PrismBridge	STDW	SWAT
Portraits	11.6	10.2	14.6
CIFAR-10	596.9	576.3	546.3
CIFAR-100	488.2	326.5	408.6
Office-31	144.7	206.3	401.6

On Office-31, the corresponding mean accuracies are 81.89% for PrismBridge, 73.58% for STDW, and 67.92% for SWAT.