LOCALLY CORRECT, GLOBALLY WRONG: TASK IDENTITY RECOVERY FOR REPLAY-FREE MULTI-SOURCE CLASS-INCREMENTAL LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Replay-free multi-source class-incremental learning requires models to learn from changing data sources without revisiting past data. Beyond forgetting, two challenges arise. First, independently trained source-specific classifiers may produce scores that are not jointly calibrated, so a global decision over all classes can be wrong even when the corresponding local classifier is correct. Second, new sources may introduce new classes or repeat existing label sets, requiring the model to decide whether to create a new task branch or adapt an existing one. We address this problem with HiDMoA (Hierarchical Decoupled Mixture-of-Adapters). The key idea is to introduce an intermediate task level, where tasks are characterized by source-specific feature distributions. These distributional representations enable two-stage classification: first recovering the input's task identity, then classifying the input within that task's label space, thereby avoiding direct comparisons of scores across independently trained source-specific classifiers. Specifically, HiDMoA uses one source fVAE (feature variational autoencoder) per source for reconstruction-based task identification, while each task branch combines task-branch adapters and a local cosine head on a frozen pretrained backbone for within-task classification. The same reconstruction evidence determines whether a source with a repeated label set joins an existing task branch or starts a new one. Reused task branches are updated through replay-free distillation. On an 8-source, 50-class industrial surface-defect stream, HiDMoA achieves 93.1%, 93.4%, and 92.6% Last Macro-F1 across three source orders, outperforming all evaluated baselines in each order. Task-identity accuracy reaches 99.72%, bringing Last Macro-F1 within one point of an oracle given the true task identity. On an 8-source, 135-class general multi-source stream, HiDMoA-ResNet-50 achieves 95.4% average Macro-F1 and leads at seven of eight evaluation steps. On a repeated-label stream, HiDMoA achieves 95.8% Last Macro-F1, compared with 85.4% for the strongest evaluated baseline.

1 INTRODUCTION

Class-incremental learning (CIL) requires a model to learn new classes sequentially while continuing to recognize all previously seen classes. Under replay-free constraints, earlier samples cannot be revisited, so updates on new classes may degrade previous performance, leading to catastrophic forgetting (Van de Ven et al., 2022; Masana et al., 2022; Rebuffi et al., 2017). Multi-source streams further complicate this setting: each source may introduce new classes alongside shifts in acquisition protocols, visual appearance, and class distributions (Lee et al., 2024). For example, in industrial defect recognition, new materials, sensors, inspection protocols, and defect types emerge over time, while retaining historical data may be costly or subject to restrictions. As these variations accumulate across sources, assigning a test input to the correct class among all previously seen classes becomes increasingly difficult.

Global prediction requires a decision over all seen classes. When this decision takes one $\arg\max$ over independently learned local-head scores, it relies on their comparability across sources. However, source-specific classifiers learned in separate sources are not jointly calibrated, and replay-free training provides no previous-source samples for aligning their scores. As a result, global compar-

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.

ison can select a class from the wrong source even when the classifier for the true source predicts correctly within its local label space, as illustrated in Figure 1. This motivates an intermediate task level that separates task identification from within-task classification. That is, we first identify the input's task branch, then apply its local cosine head within that branch's label space, avoiding direct comparisons of class scores across sources. Task identification requires not only selecting the correct task branch at test time but also establishing appropriate source-to-task assignments during training. This raises a question: when does a new source require a new task branch? The same defect classes, such as cracks and scratches, may reappear under different sensors or imaging conditions, making it difficult to determine whether the new source can share an existing task branch. Without access to earlier data, the model cannot directly compare source distributions or check whether updating that task branch preserves its performance on previous sources.

Our setting therefore requires two complementary capabilities: preserving task-local discrimination during replay-free updates and recovering the task identity of each input. Existing regularization and replay preserve old knowledge, while expansion and parameter-efficient adaptation reduce interference from new updates (Kirkpatrick et al., 2017; Rebuffi et al., 2017; Yan et al., 2021; Liang & Li, 2024). Although effective against forgetting, these approaches do not explicitly recover the missing task identity. Task-prediction methods use expert gates, OOD (out-of-distribution) responses, or likelihood ratios (Aljundi et al., 2017; Kim et al., 2022a; Lin et al., 2024; Gupta et al., 2025). For methods that infer task identity from task-specific classifier scores, comparing scores across independently learned tasks may require joint calibration or replayed samples. Generative classifiers and variational-expert models offer distribution-based alternatives (Van De Ven et al., 2021; Hihn & Braun, 2021; Solomon et al., 2024). However, generative classifiers make global predictions by comparing class-conditional likelihoods learned independently across sources, whose scales may become inconsistent under heterogeneous source shifts.Variational-expert models reduce interference through expert specialization, while global prediction requires coordination between the router and its experts. Existing routing and expansion mechanisms also support expert reuse, but source-to-task assignment must additionally determine which arriving sources can share a task-local classifier. Both training-time assignment and inference-time routing therefore depend on reliable cross-source evidence, often obtained through calibration or replay.

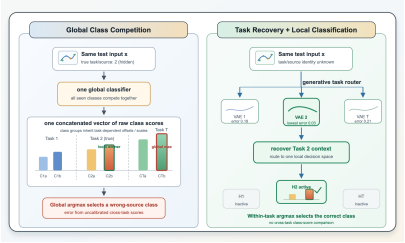


Figure 1: Global prediction vs. task-routed local prediction.

This gap motivates HiDMoA (Hierarchical Decoupled Mixture-of-Adapters), a replay-free framework that separates task-identification evidence from class-discriminative scores. It retains one source fVAE for each source and uses these fVAEs to determine which sources can share a task branch. At inference, global classification proceeds in two stages: first recovering the input's task branch, then predicting its class within that branch's label space. Specifically, each task branch comprises task-branch adapters and a local cosine head on a frozen pretrained backbone. When an arriving source is compatible with an existing task branch and has the same label set, that task branch is updated with replay-free distillation, while other task branches remain frozen. The source retains its own fVAE, allowing multiple sources to share one task branch while preserving their distinct distributional representations.

Our contributions are threefold. First, we formalize how task-identification errors contribute to global classification risk and characterize when score recalibration alone is insufficient, motivating task recovery from separate source-fVAE evidence. Second, we introduce HiDMoA, a replay-free framework that separates source-fVAE routing from task-branch classification. It retains a distinct source fVAE for each source while allowing compatible sources with the same label set to share a task branch, using reconstruction evidence for both source-to-task-branch assignment during training and task recovery at inference. A shared task branch is adapted with replay-free distillation while unmatched task branches remain frozen. Third, we demonstrate strong performance across industrial and general multi-source streams and validate the roles of task recovery and source-to-task assignment. Reconstruction routing achieves 99.7% task-identity accuracy and brings Last Macro-F1 within one point of the task-identity oracle. On the repeated-label stream, HiDMoA achieves 95.8% Last Macro-F1, compared with 85.4% for the strongest baseline. Head-calibration and routing ablations, together with controlled simulations, further examine the proposed error mechanism.

2 Related Work

Preserving old knowledge and reducing interference. CIL methods preserve earlier knowledge through regularization, replay, constrained adaptation, or additional model capacity. LwF and EWC constrain learned responses or important parameters (Li & Hoiem, 2017; Kirkpatrick et al., 2017), while OGD and GPM restrict update directions (Farajtabar et al., 2020; Saha et al., 2021). iCaRL and MRFA preserve past information by replaying stored examples or augmented features (Rebuffi et al., 2017; Zheng et al., 2024). For pretrained models, InfLoRA and SD-LoRA constrain the adaptation space to reduce interference (Liang & Li, 2024; Wu et al., 2025). Expansion and modular methods, including DER, BEEF, TagFex, and PEC, allocate task- or class-specific capacity (Yan et al., 2021; Wang et al., 2022; Zheng et al., 2025; Zając et al., 2024). These methods primarily address forgetting or interference, but do not directly recover task identity at test time.

Task-identity inference and expert models. Task-identity inference determines which task or expert should process a test input. Expert Gate selects task-specific experts using autoencoder reconstruction errors (Aljundi et al., 2017), while conditional channel gating and iTAML incorporate task inference into task-dependent computation (Abati et al., 2020; Rajasegaran et al., 2020). Expert Gate also uses reconstruction-based task relatedness to guide knowledge transfer. MORE, TPL, BUILD, and task-specific BN use OOD or likelihood-based evidence to select task-specific predictors (Kim et al., 2022a; Lin et al., 2024; Gupta et al., 2025; Zhou et al., 2026). Expert-based methods such as SEED, SEMA, CaRE, TAME, and MoTE combine specialized modules through ensembles, adaptation, or learned routing (Rypeść et al., 2024; Wang et al., 2025; Lou et al., 2026; Zhu et al., 2024; Li et al., 2025). SEMA adapts its capacity by reusing or adding adapters, and TAME reuses experts when tasks recur. Variational and hierarchical expert models learn routing together with expert specialization (Hihn & Braun, 2021; 2023). For routing based on classifier-derived scores, accurate within-task classification does not guarantee comparability across tasks; expert reuse also leaves the question of how separately modelled sources should share a task-local classifier.

Generative models for replay and prediction. Generative replay methods, including EEC, Generative Feature Replay, DDGR, GUIDE, and TaskVAE, synthesize samples or features from earlier tasks to preserve learned knowledge (Ayub & Wagner, 2021; Liu et al., 2020; Gao & Liu, 2023; Cywinski et al., 2025; Kann et al., 2025). Generative Classifier predicts labels from class-conditional likelihoods (Van De Ven et al., 2021). U-TELL uses generative task representations for expert selection (Solomon et al., 2024), while autoencoder-based incremental learning uses latent class prototypes for classification (Choi et al., 2019). However, using generative models for replay or prediction does not by itself determine whether an arriving source should form a new task branch or share an existing one.

3 Method

3.1 Problem Setting and Notation

Sources arrive one at a time as $\{(\mathcal{D}_t, \mathcal{C}_t)\}_{t=1}^{T}$, where $\mathcal{D}_t$ denotes the labeled dataset of visual source t, and $\mathcal{C}_t = \{y : (x, y) \in \mathcal{D}_t\}$ is the source label set. We split the current data into $\mathcal{D}_t = \mathcal{D}_t^{\text{train}} \cup \mathcal{D}_t^{\text{val}}$ and write $n_t = |\mathcal{D}_t^{\text{train}}|$. Source boundaries are observed during training, and previous-source samples cannot be revisited under the replay-free constraint. See Appendix A for the main notation.

At training time, the task-branch assignment of an arriving source is not given: source t may join an existing task branch or start a new one. After processing t sources, the model has $K_t \leq t$ task branches. Each branch k is a reusable module (A^k, H_k), comprising task-branch adapters A^k and a local cosine head H_k defined over the branch label set $\mathcal{Y}_k$. Let $\mathcal{S}_k$ index the sources assigned to branch k; then $\mathcal{C}_s = \mathcal{Y}_k$ for every $s \in \mathcal{S}_k$. Each source s retains a separate source fVAE V_s fitted only to that source. After learning source t, the model must predict over the cumulative label set $\mathcal{C}_{1:t} = \bigcup_{t' \leq t} \mathcal{C}_{t'}$ for a test input x without observing its task identity.

3.2 Source-to-Task Assignment

Source representation. We represent source t by the distribution of its routing features $\mathcal{R}_t = \{G(x_i) : (x_i, y_i) \in \mathcal{D}_t^{\text{train}}\}$. There is one feature per training sample, so $|\mathcal{R}_t| = n_t$. The routing

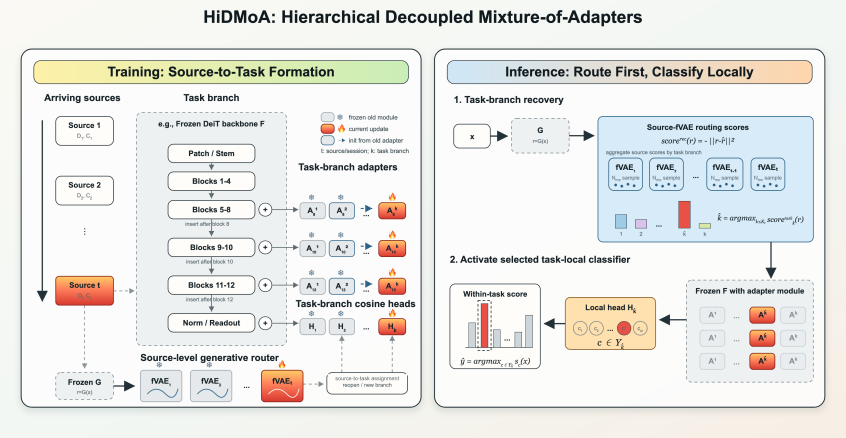


Figure 2: HiDMoA overview. Each source retains a source fVAE, while compatible sources share a task branch with adapters and a local cosine head. At inference, source-fVAE scores select the task branch for local classification.

encoder maps each input to $G(x) \in \mathbb{R}^m$ and remains fixed throughout learning. All sources are therefore modelled in the same feature space. Appendix G specifies the network used.

As shown in Figure 2, for each source t, we fit and retain one source fVAE V_t that models $\mathcal{R}_t$ using only that source's routing features. This variational autoencoder (Kingma & Welling, 2013) comprises an encoder with approximate posterior $q_t(z|r) = \mathcal{N}(\mu_t(r), \mathrm{diag}(\sigma_t^2(r)))$, a fixed standard normal prior $p(z) = \mathcal{N}(0, I)$, and a decoder with conditional distribution $p_t(r|z) = \mathcal{N}(\hat{r}_t(z), I)$. For each $r_i \in \mathcal{R}_t$, we draw one latent sample $z_i \sim q_t(z|r_i)$ using the reparameterization trick. We fit V_t by minimizing

$$\mathcal{L}_{\text{router}}^{(t)} = \frac{1}{n_t} \sum_{r_i \in \mathcal{R}_t} \left[\lambda_{\text{rec}} \frac{1}{m} \|r_i - \hat{r}_t(z_i)\|_2^2 + \beta \, \mathrm{KL}(q_t(z|r_i) \| p(z)) \right]. \tag{1}$$

The reconstruction term fits V_t to the routing features of source t, while the KL term regularizes its posterior toward the standard normal prior. We freeze V_t after fitting and use $\lambda_{\text{rec}} = 1.0$, $\beta = 0.01$.

Source fVAE score. Consistent with the reconstruction objective, the router scores a stored source fVAE V_s by how closely it reconstructs a routing feature r. The score is the negative per-dimension reconstruction error averaged over N_{mc} posterior samples,

$$\psi_s^{\text{recon}}(r) = -\frac{1}{N_{\text{mc}}} \sum_{j=1}^{N_{\text{mc}}} \frac{1}{m} \|r - \hat{r}_s(z^{(j)})\|_2^2, \qquad z^{(j)} \sim q_s(z|r), \tag{2}$$

where s indexes a stored source fVAE and a larger score indicates a closer reconstruction of r by V_s. All source fVAEs are evaluated on the same frozen representation with the same per-dimension metric, providing a common measurement basis. The default configuration uses $N_{\text{mc}} = 600$ posterior samples. For training-time decisions on a sample set D, we first average each source score over the inputs and then take the maximum across the sources assigned to each task branch. With labels omitted from the summation notation, the set-level scores are $\bar{\psi}_s^{\text{recon}}(D) = |D|^{-1} \sum_{x \in D} \psi_s^{\text{recon}}(G(x))$ and $\bar{\psi}_k^{\text{task}}(D) = \max_{s \in \mathcal{S}_k} \bar{\psi}_s^{\text{recon}}(D)$.

Source-to-task assignment. After fitting V_t, we make a single source-to-task assignment decision, before training any task-branch parameters on $\mathcal{D}_t^{\text{train}}$:

$$a_t = \begin{cases} k, & \text{source } t \text{ joins task branch } k \le K_{t-1}, \\ \text{new}, & \text{source } t \text{ starts task branch } K_{t-1} + 1. \end{cases}$$

We first restrict the candidate branches to $\mathcal{E}_t = \{k \le K_{t-1} : \mathcal{C}_t = \mathcal{Y}_k\}$. If $\mathcal{E}_t = \varnothing$, we set $a_t = \text{new}$. Otherwise, we assess each eligible branch's reconstruction of the current source relative to that of the source's own fVAE V_t. Since reconstruction scores are non-positive, we negate them to obtain

reconstruction errors and compare them on the current validation set through the ratio

$$\Delta_{\mathrm{TF}}(t,k) = \max\{-\bar{\psi}_k^{\mathrm{task}}(\mathcal{D}_t^{\mathrm{val}}), \epsilon\} \big/ \max\{-\bar{\psi}_t^{\mathrm{recon}}(\mathcal{D}_t^{\mathrm{val}}), \epsilon\}.$$

Here $\epsilon = 10^{-12}$ provides a numerical floor and prevents a vanishing denominator. Among the eligible branches, we select $\tilde{k} = \arg\min_{k \in \mathcal{E}_t} \Delta_{\mathrm{TF}}(t,k)$. We assign $a_t = \tilde{k}$ if $\Delta_{\mathrm{TF}}(t,\tilde{k}) \le \gamma_{\mathrm{TF}}$; otherwise, we set $a_t = \mathrm{new}$. The global threshold is shared across sources and defaults to $\gamma_{\mathrm{TF}} = 5$. Joining leaves the number of task branches unchanged, $K_t = K_{t-1}$, and records $\mathcal{S}_k \leftarrow \mathcal{S}_k \cup \{t\}$; starting a task branch sets $K_t = K_{t-1} + 1$ and $\mathcal{S}_{K_t} = \{t\}$.

3.3 Task-Branch Classification and Updating

Task-branch adapters. Each task branch specializes a frozen backbone $F = F_L \circ F_{L-1} \circ \cdots \circ F_1$ through residual adapters. When a new branch k is created, we allocate its adapter set $A^k = \{A_\ell^k\}_{\ell \in \mathcal{L}}$ at the selected stages $\mathcal{L}$. Let $u_{\ell-1}^{(k)}$ denote the input to stage ℓ, with $u_0^{(k)} = x$, and let $h_\ell^{(k)}(x) = F_\ell(u_{\ell-1}^{(k)}(x))$ be the backbone output at that stage. At each selected stage, HiDMoA adds the adapter output as a residual; all other stages are unchanged:

$$u_\ell^{(k)} = h_\ell^{(k)}(x) + A_\ell^k(h_\ell^{(k)}(x)) \ \ (\ell \in \mathcal{L}), \qquad u_\ell^{(k)} = h_\ell^{(k)}(x) \ \ (\ell \notin \mathcal{L}),$$

with branch-conditioned feature $\phi^{(k)}(x) = \mathsf{U}(u_L^{(k)}) \in \mathbb{R}^d$, where U is the backbone readout.

An adapter with M_ℓ local experts is a gated mixture: $A_\ell^k(h) = \sum_{j=1}^{M_\ell} g_{\ell,j}^k(h) a_{\ell,j}^k(h)$, $g_\ell^k(h) = \mathrm{softmax}(W_\ell^k \rho(h))$. Here $\rho(\cdot)$ produces the pooled feature, and W_ℓ^k parameterizes the gate within task branch k. For $M_\ell = 1$, the adapter reduces to $A_\ell^k(h) = a_{\ell,1}^k(h)$. These gates combine experts within a branch; source-fVAE evidence selects the task branch.

New task branch: local cosine head and objective. If $a_t = \mathrm{new}$, we create branch $k = K_{t-1}+1$ with $\mathcal{Y}_k = \mathcal{C}_t$ and $\mathcal{S}_k = \{t\}$. Its local cosine head H_k has one prototype $\mu_c^{(k)}$ for each class $c \in \mathcal{Y}_k$. The class score is the cosine similarity between the prototype and the branch-conditioned feature $\phi^{(k)}(x)$, scaled by a fixed τ:

$$s_c^{(k)}(x) = \tau \langle \phi^{(k)}(x), \mu_c^{(k)} \rangle \big/ \big(\|\phi^{(k)}(x)\|_2 \|\mu_c^{(k)}\|_2 \big), \qquad c \in \mathcal{Y}_k.$$

Before optimization, we initialize each prototype by class-mean imprinting on the current training data: $\mathcal{D}_{t,c}^{\mathrm{train}} = \{x_i : (x_i, y_i) \in \mathcal{D}_t^{\mathrm{train}}, y_i = c\}$, $\mu_c^{(k)} \leftarrow \mathrm{Norm}(|\mathcal{D}_{t,c}^{\mathrm{train}}|^{-1} \sum_{x_i \in \mathcal{D}_{t,c}^{\mathrm{train}}} \phi^{(k)}(x_i))$, $\mathrm{Norm}(v) = v/\|v\|_2$. Thus, each prototype starts at the normalized feature centroid of its class. The prototypes and adapters are then optimized jointly with task-local cross-entropy:

$$\mathcal{L}_{\mathrm{local}}^{(t\to k)} = -\frac{1}{|\mathcal{D}_t^{\mathrm{train}}|} \sum_{(x_i,y_i) \in \mathcal{D}_t^{\mathrm{train}}} \log \frac{\exp(s_{y_i}^{(k)}(x_i))}{\sum_{c \in \mathcal{Y}_k} \exp(s_c^{(k)}(x_i))}. \tag{3}$$

The softmax normalization is restricted to $\mathcal{Y}_k$, so each local cosine head is trained within its own label space. Task-first inference does not require a shared class-score scale across branches. Training a new branch updates only its adapter experts, their gates when $M_\ell > 1$, and its prototypes. The classification backbone F, routing encoder G, all fVAEs, and all other task branches remain frozen.

Remark: warm-starting a new task branch. The stored source fVAEs also provide an initialization signal through their available reconstruction evidence. When source $t > 1$ forms a new task branch k, we select the existing branch with the highest set-level score on the current training data: $j^*(t) = \arg\max_{j \le K_{t-1}} \bar{\psi}_j^{\mathrm{task}}(\mathcal{D}_t^{\mathrm{train}})$. The score is defined in subsection 3.2. Within branch $j^*(t)$, let $Q_e(t)$ be the average gate response of expert e over the current samples and adapted stages. We select $e^*(t) = \arg\max_e Q_e(t)$, or the sole expert when $M_\ell = 1$. The new task-branch adapter is initialized by $a_{\ell,1}^k \leftarrow a_{\ell,e^*(t)}^{j^*(t)}$ for $\ell \in \mathcal{L}$ and then fine-tuned on $\mathcal{D}_t^{\mathrm{train}}$; Appendix D compares this choice with random initialization.

Existing task branch: reopening and merge update. If $a_t = k \le K_{t-1}$, we allocate no new task branch and reopen only (A^k, H_k). All other task branches, F, G, and all source fVAEs remain frozen. Since samples from the sources previously assigned to k cannot be revisited, we use a frozen pre-update copy $(A_{\mathrm{old}}^k, H_{k,\mathrm{old}})$ as a teacher to anchor the update. Let $s^{(k)}(x) = (s_c^{(k)}(x))_{c \in \mathcal{Y}_k}$ and $s_{\mathrm{old}}^{(k)}(x)$ denote the student and teacher logits, respectively. The student combines the local classification objective with output distillation:

$$\mathcal{L}_{\mathrm{kd}} = \frac{T_{\mathrm{kd}}^2}{|\mathcal{D}_t^{\mathrm{train}}|} \sum_{x \in \mathcal{D}_t^{\mathrm{train}}} \mathrm{KL}\Big(\mathrm{softmax}(s_{\mathrm{old}}^{(k)}(x)/T_{\mathrm{kd}}) \,\Big\|\, \mathrm{softmax}(s^{(k)}(x)/T_{\mathrm{kd}})\Big),$$
$$\mathcal{L}_{\mathrm{update}} = \mathcal{L}_{\mathrm{local}}^{(t\to k)} + \lambda_{\mathrm{kd}} \mathcal{L}_{\mathrm{kd}}, \tag{4}$$

where $\mathcal{L}_{\mathrm{local}}^{(t\to k)}$ is equation 3 evaluated on $\mathcal{D}_t^{\mathrm{train}}$ using the matched local cosine head and its label set $\mathcal{Y}_k$. The classification term fits the current source, while the distillation term encourages consistency with the branch's pre-update outputs on current-source inputs. Both teacher and student logits are evaluated only on $\mathcal{D}_t^{\mathrm{train}}$, so the update remains replay-free. The default temperature is $T_{\mathrm{kd}} = 1$. For each merge, λ_{kd} is selected using the current source's validation set (Appendix G). After training, we update $\mathcal{S}_k \leftarrow \mathcal{S}_k \cup \{t\}$ and register the source fVAE V_t under branch k.

3.4 Inference: Task Identity Recovery and Local Prediction

Given a test input x after learning source t, we first compute the frozen routing feature $r = G(x)$. Every stored source fVAE receives this same feature and produces the reconstruction score $\psi_s^{\mathrm{recon}}(r)$ defined in equation 2. To recover the task identity, we take the maximum source score within each task branch and select the branch with the highest aggregate score: $\psi_k^{\mathrm{task}}(r) = \max_{s\in\mathcal{S}_k} \psi_s^{\mathrm{recon}}(r)$, $\hat{k} = \arg\max_{k\le K_t} \psi_k^{\mathrm{task}}(r)$. For local prediction, we activate only the selected task branch $(A^{\hat{k}}, H_{\hat{k}})$. Its adapted backbone produces the branch-conditioned feature, and its local cosine head predicts the class within $\mathcal{Y}_{\hat{k}}$: $\hat{y} = \arg\max_{c\in\mathcal{Y}_{\hat{k}}} s_c^{(\hat{k})}(x)$. Algorithm 1 gives the complete procedure, including source-score aggregation, task-branch selection, and local prediction.

4 Theoretical Results

We compare three prediction classes under explicit conditions and then isolate how task-branch selection affects classification risk with fixed local heads. Throughout this section, we condition on the learned models after source t and write $K = K_t$. For the hypothesis-class comparison, inference randomness is also held fixed; the risk identities remain valid when averaged over Monte Carlo randomness. Prior theory relates within-task and task-ID prediction (Kim et al., 2022b).

For branch k, let $s^{(k)}(x) = (s_c^{(k)}(x))_{c\in\mathcal{Y}_k}$ and define

$$f_k(x) = \arg\max_{c\in\mathcal{Y}_k} s_c^{(k)}(x), \qquad \xi_k(x) = \max_{c\in\mathcal{Y}_k} s_c^{(k)}(x), \qquad \boldsymbol{\xi}(x) = (\xi_1(x), \dots, \xi_K(x)).$$

Here, f_k is the local classifier and ξ_k its maximum class score. The set $\mathcal{K}^*(x) = \{k \le K : y \in \mathcal{Y}_k\}$ contains the branches whose label sets admit the true label. Define $e_{\mathrm{route}} = \Pr[\hat{k} \notin \mathcal{K}^*(x)]$ and $e_{\mathrm{min}}^{\mathrm{cal}} = \inf_\kappa \Pr[\kappa(\boldsymbol{\xi}(x)) \notin \mathcal{K}^*(x)]$, where the infimum is over measurable maps from the score vector to a branch.

The coupled rule uses the same local scores for branch selection and classification, whereas HiD-MoA selects a branch using source-fVAE evidence:

$$\hat{y}_{\mathrm{cal}} = f_{\kappa_{\mathrm{cal}}(x)}(x), \quad \kappa_{\mathrm{cal}}(x) = \arg\max_{k\le K} T_k(\xi_k(x)), \qquad \hat{y} = f_{\hat{k}}(x), \quad \hat{k} = \arg\max_{k\le K} \psi_k^{\mathrm{task}}(G(x)).$$

Each T_k is strictly increasing and therefore preserves within-branch rankings. Ties are resolved by fixed, consistent rules. When label ownership is unique, write $\mathcal{K}^*(x) = \{k^*\}$ and define $\bar{\varepsilon} = \Pr[f_{k^*}(x) \ne y]$ and $\mathcal{O} = \{f_{k^*}(x) = y\}$. These describe the error and correct-prediction event of a task-aware oracle using the same local heads.

Definition 1 (Three prediction classes)**.** Let Φ be a class of feature maps $\mathcal{X} \to \mathbb{R}^d$ and $\mathcal{R}$ a class of routers $\mathcal{X} \to \{1, \dots, K\}$. Write $R(h) = \Pr[h(x) \ne y]$, let R^* be the Bayes classification risk, and define $R(\mathcal{H}) = \inf_{h\in\mathcal{H}} R(h)$. With the local heads held fixed, the three classes are

$$\mathcal{H}_{\mathrm{joint}} = \Big\{x \mapsto \arg\max_{c\in\bigcup_k \mathcal{Y}_k} (\langle w_c, \phi(x)\rangle + \beta_c) : \phi \in \Phi,\ w_c \in \mathbb{R}^d,\ \beta_c \in \mathbb{R}\Big\},$$
$$\mathcal{H}_{\mathrm{cpl}} = \big\{x \mapsto f_{\kappa_{\mathrm{cal}}(x)}(x) : T_k(u) = a_k u + b_k,\ a_k > 0\big\},$$
$$\mathcal{H}_{\mathrm{dec}} = \big\{x \mapsto f_{q(x)}(x) : q \in \mathcal{R}\big\}.$$

The joint class learns a global head and a feature map; the coupled class calibrates fixed local heads; the decoupled class varies a separate router. The intercepts β_c allow the joint class to represent affine

score calibration when the additional common-feature condition (A7) holds. We use $\mathrm{Comp}(\mathcal{H})$ for the disagreement covering entropy defined in Appendix B.

Proposition 1 compares the approximation risk and complexity of these three classes under (A1)–(A6) in Appendix B, relating the score-statistic risk floor to the flexibility of separate routing.

Proposition 1 (Three classes). *Under (A1)–(A6) in Appendix B,*

$$\begin{aligned} R(\mathcal{H}_{\mathrm{cpl}}) - R^* \;\geq\; e^{\mathrm{cal}}_{\min} - R^* \;\geq\; R(\mathcal{H}_{\mathrm{dec}}) - R^*, \\ \mathrm{Comp}(\mathcal{H}_{\mathrm{cpl}}) \;\leq\; \mathrm{Comp}(\mathcal{H}_{\mathrm{dec}}) \;\leq\; \mathrm{Comp}(\mathcal{H}_{\mathrm{joint}}). \end{aligned} \tag{5}$$

If also (A7) holds, then $R(\mathcal{H}_{\mathrm{joint}}) \leq R(\mathcal{H}_{\mathrm{cpl}})$.

Coupled prediction retains the risk floor induced by $\boldsymbol{\xi}(x)$, while affine calibration adjusts only $2K$ parameters. A joint model can avoid this statistic-specific floor through learned features and global class comparisons, with a parameter budget that grows with the class vocabulary and feature-map complexity. The proposed class sits between the two under (A5)–(A6): separate routing provides evidence beyond $\boldsymbol{\xi}$, with adjustable complexity determined by $\mathcal{R}$.

Proposition 2 (Fixed heads). *Under (A1)–(A4), both predictors use the same fixed local heads, so their oracle error* $\bar{\varepsilon}$ *and event* $\mathcal{O}$ *coincide. Consequently,*

$$\Pr[\hat{y} \neq y] = \bar{\varepsilon} + \Pr[\hat{k} \neq k^*,\, \mathcal{O}] \qquad \Pr[\hat{y}_{\mathrm{cal}} \neq y] \;\; = \bar{\varepsilon} + \Pr[\kappa_{\mathrm{cal}} \neq k^*,\, \mathcal{O}] \;\geq\; e^{\mathrm{cal}}_{\min}. \tag{6}$$

Subtracting and cancelling $\{\kappa_{\mathrm{cal}} \neq k^*,\, \hat{k} \neq k^*,\, \mathcal{O}\}$ *gives*

$$\Pr[\hat{y}_{\mathrm{cal}} \neq y] - \Pr[\hat{y} \neq y] = \Pr[\kappa_{\mathrm{cal}} \neq \hat{k},\, \hat{k} = k^*,\, \mathcal{O}] - \Pr[\kappa_{\mathrm{cal}} \neq \hat{k},\, \kappa_{\mathrm{cal}} = k^*,\, \mathcal{O}]. \tag{7}$$

Proposition 2 isolates the effect of branch selection while keeping within-branch classifiers fixed. The risk difference depends on inputs in $\mathcal{O}$ for which exactly one rule selects the correct branch. Thus, decoupled prediction has lower risk precisely when its probability of correct routing on $\mathcal{O}$ exceeds that of the coupled rule. A large local class score alone does not guarantee correct task identification. The proofs appear in Appendices B.1 and B.2. When labels overlap, the general identity $\Pr[\hat{y} \neq y] = e_{\mathrm{route}} + \varepsilon_{\mathrm{local}}$ still holds, where $\varepsilon_{\mathrm{local}}$ is the joint probability of selecting a label-admissible branch and misclassifying the input (equation 9). Its class-balanced form applies to mean class recall, not directly to Macro-F1. Appendix B.3 further gives sufficient reconstruction-margin and Monte Carlo accuracy conditions for routing.

5 Experiments

5.1 Experimental Setup

Datasets and evaluation protocol. We evaluate multi-source class-incremental learning on two eight-source streams. The primary benchmark contains 50 classes for industrial surface recognition from `neu`, `xsdd`, `magnetic`, `cr7`, `dagm`, `gc10`, `kolek`, and `bsd` (Song & Yan, 2013; Feng et al., 2021; Huang et al., 2020; Jiang & Wang, 2024; Wieler et al., 2007; Lv et al., 2020; Tabernik et al., 2020; Schlagenhauf & Landwehr, 2021). The general benchmark contains 135 classes from `cifar10`, `deepweeds`, `eurosat`, `fashion_mnist`, `mnist`, `oxford_pet`, `patternnet`, and `food11` (Krizhevsky et al., 2009; Olsen et al., 2019; Helber et al., 2019; Xiao et al., 2017; LeCun et al., 1998; Parkhi et al., 2012; Zhou et al., 2018; Singla et al., 2016). Both benchmarks use disjoint source label sets, with one source per task branch and no task identity provided at test time. After each incremental step, we evaluate on the test data from all sources seen so far. Appendix F defines Macro-F1 and mAcc and summarizes the dataset statistics, image resolution, and data splits. Results on the general multi-source benchmark are reported in Appendix F.2.

Compared methods. The baselines cover three approaches: task-identity inference or OOD-based task selection (BUILD (Gupta et al., 2025), MORE (Kim et al., 2022a), and TPL (Lin et al., 2024)); generative evidence (Generative Classifier (Van De Ven et al., 2021)); and modular or expert-based expansion (DER (Yan et al., 2021), PEC (Zając et al., 2024), SEED (Rypeść et al., 2024), SEMA (Wang et al., 2025), and TagFex (Zheng et al., 2025)). All methods use the same source orders, data splits, image resolution, and random seeds. Method-specific optimization and replay settings are reported in Appendix G. Appendix H reports the efficiency and resource usage of the methods.

5.2 Main Results

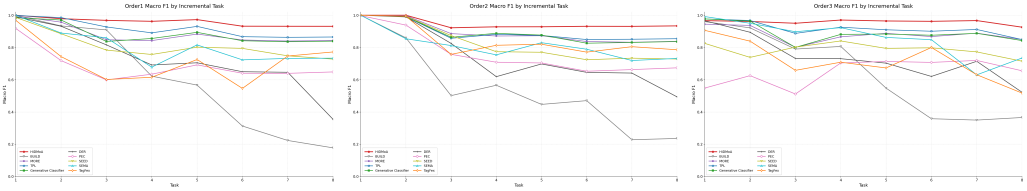
On the industrial surface-defect stream, Table 1 separates baselines with explicit task-identity inference or OOD-based task selection from those that predict directly over all seen classes. HiDMoA achieves the highest Last and Avg Macro-F1 in all three source orders; the corresponding mAcc results appear in Table 11. Averaged across the three orders, Last Macro-F1 is 93.0% for HiDMoA, 85.6% for TPL, and 84.4% for MORE, the closest task-identity baselines. The order-specific Last scores range from 92.6% to 93.4%, a difference of 0.8 percentage points.

Table 1: Industrial surface-defect stream with 8 sources and 50 classes. Last reports Macro-F1 after the final source; Avg averages cumulative Macro-F1 over incremental steps. The first baseline block uses explicit task-identity inference or OOD-based task selection; the second predicts directly over all seen classes. Values are mean ± standard deviation over 3 seeds; best result is bold.

Method	Order 1		Order 2		Order 3	
	Last Macro-F1	Avg Macro-F1	Last Macro-F1	Avg Macro-F1	Last Macro-F1	Avg Macro-F1
BUILD	17.8±1.50	59.3±4.90	23.6±3.20	53.9±10.40	36.6±4.70	63.8±0.50
MORE	84.4±1.50	88.2±1.30	83.8±0.50	89.2±0.50	85.0±0.40	88.0±0.70
TPL	86.5±0.60	91.6±0.50	85.4±0.40	89.5±0.30	84.9±0.70	91.6±0.40
Generative Classifier	83.9±0.50	88.2±0.50	83.7±0.30	88.9±0.50	84.3±0.40	88.9±0.50
DER	35.5±9.60	72.3±0.60	49.4±8.80	73.9±2.70	52.2±6.00	73.5±1.00
PEC	64.9±1.40	68.6±0.60	67.4±1.90	76.2±2.00	65.6±2.30	64.8±2.10
SEED	72.8±0.70	81.1±0.50	72.8±0.60	82.5±0.40	71.5±0.30	78.6±1.50
SEMA	73.4±3.40	80.4±2.70	73.2±4.10	81.1±7.70	73.5±2.10	85.5±0.80
TagFex	77.2±0.20	71.5±0.70	78.6±1.40	84.4±0.60	51.7±15.80	71.7±7.80
HiDMoA	**93.1±0.40**	**96.0±0.20**	**93.4±0.20**	**94.7±0.40**	**92.6±0.60**	**95.9±0.30**

The stage-wise curves in Figure 3 show performance as sources accumulate. BUILD and DER exhibit pronounced declines, whose timing and magnitude vary across source orders. Seed-to-seed variability in Table 1 also depends on source order. MORE and TPL retain explicit OOD-based task-selection signals and are the closest competitors, trailing HiDMoA by 7.4–8.6 Macro-F1 percentage points in Last performance averaged across the three orders. TagFex, SEMA, SEED, and PEC retain higher final scores than BUILD but remain below HiDMoA. The declines of several baselines highlight the difficulty of cumulative prediction once the label space spans all eight sources.

The 0.8-percentage-point range in HiDMoA's Last Macro-F1 indicates limited sensitivity to the three tested arrival orders. This observation is consistent with its decoupled inference design: classification is restricted to the selected task branch, while the router compares source-level reconstruction evidence in a shared frozen feature space.



Source order 1 Source order 2 Source order 3

Figure 3: Cumulative Macro-F1 after each incremental step on the industrial surface-defect stream.

5.3 Ablations and Repeated-Label Evaluation

Detailed connections between the three experiments below and the theory are provided in subsection D.7. Additional ablation studies are reported in Appendix D. The controlled Gaussian stream in Appendix E further separates task-identity errors from within-task classification errors.

5.3.1 Head Calibration without Test-Time Task Identity

We first assess head calibration as an alternative to explicit task routing. Joint training provides a non-incremental, all-class reference. The second and third rows compare two rules without test-time task identity: the uncalibrated rule takes the global maximum over local-head scores, whereas

task-wise affine calibration applies a learned affine correction to each head before the same global decision. The final row uses HiDMoA to identify the task branch and predict within its local head.

In Table 2, the global maximum obtains 75.65 Last Macro-F1. Task-wise affine calibration, fitted with 600 calibration samples, increases this score to 86.53. This improvement indicates that cross-head score alignment affects global prediction. It remains 5.94 Macro-F1 points below HiDMoA, showing that this fitted calibration rule doesn't match source-fVAE routing under the reported protocol.

Table 2: Head-decision rules on industrial source order 1. At test time, the three incremental settings do not receive the true task identity. HiDMoA uses randomly initialized experts.

Inference setting	Last Macro-F1	Avg Macro-F1
Joint training	93.09 ± 0.27	–
Global maximum over local heads	75.65 ± 2.15	84.04 ± 2.88
Task-wise affine calibration	86.53 ± 1.35	91.59 ± 0.28
HiDMoA, source-fVAE routing	**92.47** ± 0.44	**95.39** ± 0.23

5.3.2 Task-Identity Inference Strategy

We next compare different routing strategies. Each source corresponds to one task in this experiment, so each task branch has a single source fVAE. All fVAE-based routers use 600 Monte Carlo samples for scoring; the variants differ primarily in their fitted distribution models and scoring rules.

In Figure 4, reconstruction-based source-fVAE routing reaches 99.72 Last task-ID accuracy and 92.47 Last Macro-F1, within one point of the oracle supplied with true task identity. Source-fVAE IS routing reaches 96.92 Last task-ID accuracy, while feature prototypes and class-level fVAEs are lower. At the displayed precision, Recon and Recon+KL give essentially the same task-ID accuracies and similar Last Macro-F1, so adding KL to the routing score shows no clear benefit in this comparison.

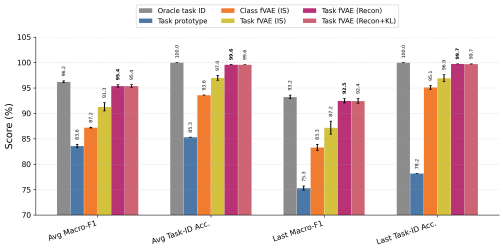


Figure 4: Task-identity inference strategy ablation. IS and Recon+KL scores are defined in Appendix D.1. "Task fVAE" denotes the source fVAE of each one-source task branch.

5.3.3 Source-to-Task Assignment on Repeated-Label Streams

To test whether a source with a repeated label set should join an existing task branch or start a new one, we split `neu`, `xsdd`, `magnetic`, and `cr7` into two class-stratified halves. This creates eight sources with four repeated label sets. All methods are evaluated in a unified label space using semantic class prediction; CN-DPM (Lee et al., 2020) is included as a task-free baseline.

The reported summary for the repeated-label stream averages over three source orders and three seeds (9 runs). HiDMoA achieves 95.80 ± 0.85 Last Macro-F1 and 95.26 ± 1.64 Avg Macro-F1; Figure 5 compares the methods. MORE, the strongest evaluated baseline, obtains 85.41 ± 1.73 and 87.64 ± 5.58.

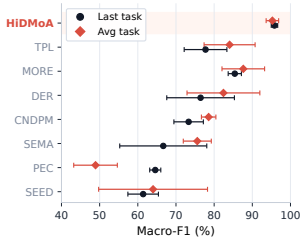


Figure 5: Macro-F1 on the repeated-label stream.

6 Conclusion

This work studies replay-free multi-source class-incremental learning when task identity is unavailable at test time. HiDMoA decouples task identity recovery from task-local classification. It aggregates source-fVAE evidence to select the task branch that best explains the input, then applies that branch's local cosine head within its label space. Across three industrial surface-defect source orders, HiDMoA achieves the strongest final and average class-balanced performance among the evaluated methods; its ResNet-50 variant also leads on the general multi-source benchmark. The routing ablations highlight the role of modelling source feature distributions in task identity recovery. The repeated-label study further evaluates source-to-task assignment, in which compatible sources with identical label sets can share an existing task branch.

AI USE STATEMENT

AI tools were used only for language polishing and presentation refinement. They were not used to design the method, derive the technical claims, implement the experiments, or generate the reported results. All scientific decisions, analyses, and final content were reviewed and controlled by the authors.

ETHICS STATEMENT

This work studies class-incremental learning for image recognition under multi-source data streams. The experiments are conducted on public or benchmark-style image datasets and do not involve human subjects, personal data, or sensitive attributes. We therefore do not identify notable ethical risks beyond the general need to validate deployed recognition systems under the data distribution and operating conditions where they will be used.

REPRODUCIBILITY STATEMENT

The paper reports the dataset splits, training protocol, model components, hyperparameters, and evaluation metrics used in the experiments. To support independent reproduction under double-blind review, we can provide an anonymized code package containing the data-preparation scripts, split files, training and routing-evaluation code, configuration files, and table-reproduction scripts. The package is intended to reproduce the reported benchmark construction, ablations, and resource measurements without requiring access to private data or author-identifying information.

REFERENCES

Davide Abati, Jakub Tomczak, Tijmen Blankevoort, Simone Calderara, Rita Cucchiara, and Babak Ehteshami Bejnordi. Conditional channel gated networks for task-aware continual learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3930–3939. IEEE, 2020.

Rahaf Aljundi, Punarjay Chakravarty, and Tinne Tuytelaars. Expert gate: Lifelong learning with a network of experts. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7120–7129. IEEE, 2017.

Ali Ayub and Alan R Wagner. EEC: Learning to encode and regenerate images for continual learning. 2021.

Euntae Choi, Kyungmi Lee, and Kiyoung Choi. Autoencoder-based incremental class learning without retraining on old data. *arXiv preprint arXiv:1907.07872*, 2019.

Bartosz Cywinski, Kamil Deja, Tomasz Trzciński, Bartłomiej Twardowski, and Łukasz Kucinski. GUIDE: Guidance-based incremental learning with diffusion models. In *ECAI 2025: 28th European Conference on Artificial Intelligence 25-30 October 2025, Bologna, Italy-Including 14th Conference on Prestigious Applications of Intelligent Systems (PAIS 2025) Proceedings*, pp. 3614–3621. SAGE Publications 1 Oliver's Yard, 55 City Road, London, EC1Y 1SP, 2025.

Mehrdad Farajtabar, Navid Azizan, Alex Mott, and Ang Li. Orthogonal gradient descent for continual learning. In *International conference on artificial intelligence and statistics*, pp. 3762–3773. PMLR, 2020.

Xinglong Feng, Xianwen Gao, and Ling Luo. X-SDD: A new benchmark for hot rolled steel strip surface defects detection. *Symmetry*, 13(4):706, 2021.

Rui Gao and Weiwei Liu. DDGR: Continual learning with deep diffusion-based generative replay. In *International Conference on Machine Learning*, pp. 10744–10763. PMLR, 2023.

Srishti Gupta, Daniele Angioni, Maura Pintor, Ambra Demontis, Lea Schönherr, Fabio Roli, and Battista Biggio. Buffer-free class-incremental learning with out-of-distribution detection. *Pattern Recognition*, pp. 112441, 2025.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. EuroSAT: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226, 2019.

Heinke Hihn and Daniel A Braun. Mixture-of-variational-experts for continual learning. *arXiv preprint arXiv:2110.12667*, 2021.

Heinke Hihn and Daniel A Braun. Hierarchically structured task-agnostic continual learning. *Machine Learning*, 112(2):655–686, 2023.

Yibin Huang, Congying Qiu, and Kui Yuan. Surface defect saliency of magnetic tile. *The Visual Computer*, 36(1):85–96, 2020.

Shengquan Jiang and Xiaoyu Wang. CR7-DET, 2024. URL `https://zenodo.org/records/14264153`. Cold-rolled steel surface defect detection dataset.

Bonpagna Kann, Sandra Castellanos-Paez, Romain Rombourg, and Philippe Lalanda. TaskVAE: Task-specific variational autoencoders for exemplar generation in continual learning for human activity recognition. In *Principle and Practice of Data and Knowledge Acquisition Workshop*, pp. 17–32. Springer, 2025.

Gyuhak Kim, Bing Liu, and Zixuan Ke. A multi-head model for continual learning via out-of-distribution replay. In *Conference on lifelong learning agents*, pp. 548–563. PMLR, 2022a.

Gyuhak Kim, Changnan Xiao, Tatsuya Konishi, Zixuan Ke, and Bing Liu. A theoretical study on solving continual learning. In *Advances in Neural Information Processing Systems*, volume 35, 2022b.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. 2013.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, 2009.

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.

Kamin Lee, Hyoeun Kim, Geunjae Choi, Hyejeong Jeon, and Nojun Kwak. Class incremental learning with large domain shift. *IEEE Access*, 12:183564–183580, 2024.

Soochan Lee, Junsoo Ha, Dongsu Zhang, and Gunhee Kim. A neural dirichlet process mixture model for task-free continual learning. In *International Conference on Learning Representations*, 2020.

Linjie Li, Zhenyu Wu, and Yang Ji. MoTE: Mixture of task-specific experts for pre-trained model-based class-incremental learning. *Knowledge-Based Systems*, 324:113795, 2025.

Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.

Yan-Shuo Liang and Wu-Jun Li. InfLoRA: Interference-free low-rank adaptation for continual learning. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23638–23647. IEEE, 2024.

Haowei Lin, Yijia Shao, Weinan Qian, Ningxin Pan, Yiduo Guo, and Bing Liu. Class incremental learning via likelihood ratio based task prediction. In *International Conference on Learning Representations*, volume 2024, pp. 50989–51028, 2024.

Xialei Liu, Chenshen Wu, Mikel Menta, Luis Herranz, Bogdan Raducanu, Andrew D Bagdanov, Shangling Jui, and Joost Van de Weijer. Generative feature replay for class-incremental learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 915–924. IEEE, 2020.

Meng Lou, Yunxiang Fu, and Yizhou Yu. Scaling continual learning to 300+ tasks with bi-level routing mixture-of-experts. *arXiv preprint arXiv:2602.03473*, 2026.

Xiaoming Lv, Fajie Duan, Jia-jia Jiang, Xiao Fu, and Lin Gan. Deep metallic surface defect detection: The new benchmark and detection network. *Sensors*, 20(6):1562, 2020.

Marc Masana, Xialei Liu, Bartłomiej Twardowski, Mikel Menta, Andrew D Bagdanov, and Joost Van De Weijer. Class-incremental learning: Survey and performance evaluation on image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5):5513–5533, 2022.

Alex Olsen, Dmitry A Konovalov, Bronson Philippa, Peter Ridd, Jake C Wood, Jamie Johns, Wesley Banks, Benjamin Girgenti, Owen Kenny, James Whinney, et al. DeepWeeds: A multiclass weed species image dataset for deep learning. *Scientific reports*, 9(1):2058, 2019.

Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pp. 3498–3505. IEEE, 2012.

Jathushan Rajasegaran, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Mubarak Shah. iTAML: An incremental task-agnostic meta-learning approach. 2020.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. iCaRL: Incremental classifier and representation learning. In *2017 IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 5533–5542. IEEE, 2017.

Grzegorz Rypeść, Sebastian Cygert, Valeriya Khan, Tomasz Trzcinski, Bartosz Zieliński, and Bartłomiej Twardowski. Divide and not forget: Ensemble of selectively trained experts in continual learning. In *International conference on learning representations*, volume 2024, pp. 22207–22224, 2024.

Gobinda Saha, Isha Garg, and Kaushik Roy. Gradient projection memory for continual learning. 2021.

Tobias Schlagenhauf and Magnus Landwehr. Industrial machine tool component surface defect dataset. *Data in Brief*, 39:107643, 2021.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

Ashutosh Singla, Lin Yuan, and Touradj Ebrahimi. Food/non-food image classification and food categorization using pre-trained GoogLeNet model. In *Proceedings of the 2nd International Workshop on Multimedia Assisted Dietary Management*. ACM, 2016.

Indu Solomon, Aye Phy Phyu Aung, Uttam Kumar, and Senthilnath Jayavelu. U-TELL: Unsupervised task expert lifelong learning. In *2024 IEEE International Conference on Image Processing (ICIP)*, pp. 1057–1063. IEEE, 2024.

Kechen Song and Yunhui Yan. A noise robust method based on completed local binary patterns for hot-rolled steel strip surface defects. *Applied Surface Science*, 285:858–864, 2013.

Domen Tabernik, Samo Šela, Jure Skvarč, and Danijel Skočaj. Segmentation-based deep-learning approach for surface-defect detection. *Journal of Intelligent Manufacturing*, 31(3):759–776, 2020.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pp. 10347–10357. PMLR, 2021.

Gido M Van De Ven, Zhe Li, and Andreas S Tolias. Class-incremental learning with generative classifiers. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 3606–3615. IEEE, 2021.

Gido M Van de Ven, Tinne Tuytelaars, and Andreas S Tolias. Three types of incremental learning. *Nature Machine Intelligence*, 4(12):1185–1197, 2022.

Fu-Yun Wang, Da-Wei Zhou, Liu Liu, Han-Jia Ye, Yatao Bian, De-Chuan Zhan, and Peilin Zhao. BEEF: Bi-compatible class-incremental learning via energy-based expansion and fusion. In *The eleventh international conference on learning representations*, 2022.

Huiyi Wang, Haodong Lu, Lina Yao, and Dong Gong. Self-expansion of pre-trained models with mixture of adapters for continual learning. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10087–10098. IEEE, 2025.

Matthias Wieler, Tobias Hahn, and Fred A Hamprecht. Weakly supervised learning for industrial optical inspection, 2007.

Yichen Wu, Hongming Piao, Long-Kai Huang, Renzhen Wang, Wanhua Li, Hanspeter Pfister, Deyu Meng, Kede Ma, and Ying Wei. SD-LoRA: Scalable decoupled low-rank adaptation for class incremental learning. 2025:58337–58353, 2025.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Shipeng Yan, Jiangwei Xie, and Xuming He. DER: Dynamically expandable representation for class incremental learning. In *2021 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 3013–3022. IEEE, 2021.

Michał Zając, Tinne Tuytelaars, and Gido van de Ven. Prediction error-based classification for class-incremental learning. In *International Conference on Learning Representations*, volume 2024, pp. 8239–8267, 2024.

Bowen Zheng, Da-Wei Zhou, Han-Jia Ye, and De-Chuan Zhan. Multi-layer rehearsal feature augmentation for class-incremental learning. In *Forty-first international conference on machine learning*, 2024.

Bowen Zheng, Da-Wei Zhou, Han-Jia Ye, and De-Chuan Zhan. Task-agnostic guided feature expansion for class-incremental learning. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10099–10109. IEEE, 2025.

Weixun Zhou, Shawn Newsam, Congmin Li, and Zhenfeng Shao. PatternNet: A benchmark dataset for performance evaluation of remote sensing image retrieval. *ISPRS journal of photogrammetry and remote sensing*, 145:197–209, 2018.

Zhiping Zhou, Xuchen Xie, Yiqiao Qiu, Run Lin, Wei-Shi Zheng, and Ruixuan Wang. Class incremental learning with task-specific batch normalization and out-of-distribution detection. *Neurocomputing*, pp. 133321, 2026.

Haoran Zhu, Maryam Majzoubi, Arihant Jain, and Anna Choromanska. TAME: Task agnostic continual learning using multiple experts. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 4139–4148. IEEE, 2024.

A NOTATION

Table 3 collects the symbols used in the paper. Two index families run throughout: t counts arriving sources, each of which contributes one source fVAE, while k counts formed task branches, each of which owns task-branch adapters and a local cosine head over its own label set.

Table 3: Symbols used in the paper, grouped by role.

Symbol	Meaning
Stream and label spaces	
T, t	number of sources; index of the arriving source
$\mathcal{D}_t$, n_t	current-source train/validation data and number of training samples, $n_t = \lvert\mathcal{D}_t^{\mathrm{train}}\rvert$
$\mathcal{R}_t$	frozen routing features from the training split of source t
$\mathcal{D}_t^{\mathrm{train}}$, $\mathcal{D}_t^{\mathrm{val}}$	train and validation splits of source t
$\mathcal{C}_t$	label set of source t
$\mathcal{C}_{1:t}$	all classes observed up to source t
Task branches	
k, K_t, K_T	task-branch index; number of task branches after source t; final task-branch count
$\mathcal{Y}_k$	label set of task branch k
$\mathcal{S}_k$	sources assigned to task branch k
a_t	assignment of source t, either a task-branch index or new
Classification modules	
F, L, $\mathcal{L}$	frozen backbone, number of stages, and adapted-stage index set
A^k, $a_{\ell,j}^k$, M_ℓ	task-branch adapters, expert j at stage ℓ, experts per stage
g_ℓ^k, W_ℓ^k, ρ	local gate, gate weight matrix, pooling function
$j^*(t)$, $e^*(t)$, $Q_e(t)$	related previous task branch, selected expert, and expert gate response
$\phi^{(k)}(x)$, d	task-branch-conditioned feature and its dimension
H_k, $\mu_c^{(k)}$, τ	local cosine head, class prototype, cosine scale
U	backbone readout producing $\phi^{(k)}(x)$
$s_c^{(k)}(x)$	score of class c under task branch k
Routing modules	
G, $r = G(x)$, m	frozen routing encoder, routing feature, its dimension
V_t, q_t, $\hat{r}_t$	source fVAE of source t, its approximate posterior and decoder mean
N_{mc}	number of posterior samples per source-fVAE score
ψ_s^q, $\psi_k^{\mathrm{task},q}$	source-fVAE routing score and its task-branch aggregate, rule q
$\bar{\psi}_s^{\mathrm{recon}}$, $\bar{\psi}_k^{\mathrm{task}}$	sample-averaged source score; maximum source-average score within a task branch
$\hat{k}$, $\hat{y}$	selected task branch and predicted class
Objectives and hyperparameters	
$\mathcal{L}_{\mathrm{router}}^{(t)}$, λ_{rec}, β	fVAE objective and its two weights
$\mathcal{L}_{\mathrm{local}}^{(t\to k)}$	task-branch cross-entropy of branch k on source t
$\mathcal{L}_{\mathrm{kd}}$, $\mathcal{L}_{\mathrm{update}}$, λ_{kd}, T_{kd}	merge distillation term, update objective, its weight and temperature
Δ_{TF}, γ_{TF}, $\mathcal{E}_t$	assignment ratio, its threshold, eligible task branches
ϵ	numerical floor in the Task Formation ratio
τ_{sim}, θ	old-expert reuse temperature and selection threshold
Analysis	
$k^*(x)$, $\kappa_{\mathrm{lab}}(\hat{y})$, f_k	unique admissible branch under (A2); branch owning the predicted label; local classifier of branch k
$s^{(k)}$, κ_{cal}	score vector of branch k; implied router $\arg\max_k T_k(\xi_k)$
e_{route}, $\bar{\varepsilon}$	probability of selecting a label-inadmissible branch; oracle local error under unique ownership
b_k, $M_{k^*\to k'}$	branch score offset; score margin against branch k'
$\mathcal{K}^*(x)$	branches owning the label of x; a singleton under disjoint label sets
$\varepsilon_{\mathrm{local}}$	joint probability of an admissible route and a local classification error
$\xi_k(x)$, $\boldsymbol{\xi}(x)$	top score of branch k; the vector of branch-wise top scores
T_k, $e_{\min}^{\mathrm{cal}}$	per-branch monotone recalibration; Bayes routing error of $\boldsymbol{\xi}(x)$
$\mathcal{H}_{\mathrm{joint}}$, $\mathcal{H}_{\mathrm{cpl}}$, $\mathcal{H}_{\mathrm{dec}}$	global, coupled, and proposed classes in Definition 1
$R(h)$, $R(\mathcal{H})$, R^*	risk $\Pr[h \neq y]$; best risk in a class; Bayes risk

Continued on next page

Symbol	Meaning
$\mathrm{Comp}(\mathcal{H})$	disagreement covering entropy at a fixed scale; distinct from the loss-class Rademacher complexity in equation 8
$\Gamma_s(r)$, osc_Γ	source-specific difference of prior and posterior divergences in equation 10; its spread across sources
γ_{recon}, $\delta_{N_{\mathrm{mc}}}$, Ω_{rec}	reconstruction margin; Monte Carlo error; bound on normalized squared reconstruction error

In Appendix B.3, Ψ_s^{recon} denotes the posterior-expected reconstruction score and $\widehat{\psi}_{s,N_{\mathrm{mc}}}^{\mathrm{recon}}$ its Monte Carlo estimate, which is the score ψ_s^{recon} used in the method. γ_{task} compares the best admissible and inadmissible population task-branch scores.

B Theoretical Analysis

This appendix proves the results in Section 4 and states their scope. We first give the conditional comparison of the three prediction classes, then derive classification-risk identities for both disjoint and overlapping label sets. We next characterize the information limit of routing from fixed local-head maxima and give a stylized additive-offset example. Finally, we analyze reconstruction-based routing under explicit separation and sampling conditions.

Setup and notation. The stream contains K task branches. Branch k has label set $\mathcal{Y}_k$ and a score vector $s^{(k)}(x) = (s_c^{(k)}(x))_{c\in\mathcal{Y}_k}$. The local classifier and the branch-wise maximum used throughout are

$$f_k(x) = \arg\max_{c\in\mathcal{Y}_k} s_c^{(k)}(x), \qquad \xi_k(x) = \max_{c\in\mathcal{Y}_k} s_c^{(k)}(x).$$

For a test input x with label y, let a router q predict $\hat{k} = q(x)$.

The decoupled predictor is

$$\hat{y} = f_{\hat{k}}(x).$$

Collect the branches that own the label of x in the admissible set

$$\mathcal{K}^*(x) = \{k \le K : y \in \mathcal{Y}_k\},$$

and call a route label-admissible when $\hat{k} \in \mathcal{K}^*(x)$. Here $\mathcal{K}^*(x)$ abbreviates a set depending on both x and its true label y. With overlapping label sets, admissibility need not recover the branch assigned to the true source. Define the label-admissibility routing error and the joint probability of a local error after an admissible route as

$$e_{\mathrm{route}} = \Pr[\hat{k} \notin \mathcal{K}^*(x)], \qquad \varepsilon_{\mathrm{local}} = \Pr[\hat{k} \in \mathcal{K}^*(x),\ f_{\hat{k}}(x) \neq y].$$

When $\mathcal{K}^*(x)$ is a singleton we write $k^* = k^*(x)$ for its element, define the task-local error

$$\varepsilon_k = \Pr[f_k(x) \neq y \mid k^* = k],$$

the task prior $\pi_k = \Pr[k^* = k]$, and the mean task-local error

$$\bar{\varepsilon} = \sum_{k=1}^{K} \pi_k \varepsilon_k = \Pr[f_{k^*}(x) \neq y].$$

Assumption 1 (Stream structure)**.** *(A1) Well-posed test:* $\mathcal{K}^*(x) \neq \varnothing$ almost surely. *(A2) Disjoint ownership:* branch label sets are pairwise disjoint, $\mathcal{Y}_k \cap \mathcal{Y}_{k'} = \varnothing$ for $k \neq k'$, so each test label has a unique admissible branch. *(A3) Consistent decisions:* ties are resolved by a fixed branch order followed by a fixed within-branch class order, consistently between the global decision over branch–class pairs and the corresponding two-stage decision. Results using strict score inequalities additionally assume that the relevant maxima have no ties.

Assumption 2 (Three-class comparison)**.** *(A4) Fixed heads:* the same scores $\{s^{(k)}\}$ are held fixed in $\mathcal{H}_{\mathrm{cpl}}$ and $\mathcal{H}_{\mathrm{dec}}$. Calibration is affine with positive slope in $\mathcal{H}_{\mathrm{cpl}}$; the fixed-head identities also hold for any fixed strictly increasing T_k. *(A5) Sufficient routing information:* writing

$$\eta_{\mathcal{R}} = \inf_{q \in \mathcal{R}} \Pr[q(x) \neq k^*,\, \mathcal{O}],$$

assume $\bar{\varepsilon} + \eta_{\mathcal{R}} \leq e^{\mathrm{cal}}_{\min}$. This is an additional sufficient condition, not a consequence of using fVAEs. *(A6) Complexity order:* at the common scale and distribution defined below,

$$\mathrm{Comp}(\mathcal{H}_{\mathrm{cpl}}) \leq \mathrm{Comp}(\mathcal{R}) \leq \mathrm{Comp}(\mathcal{H}_{\mathrm{joint}}).$$

This ordering must be established for the chosen classes if it is invoked. *(A7) Common-feature representation:* there exists $\phi_0 \in \Phi$ such that $s^{(k)}_c(x) = \langle w_c, \phi_0(x) \rangle + \beta_c$ for every k and $c \in \mathcal{Y}_k$. This is not assumed for the actual branch-conditioned features of HiDMoA.

Scope of the assumptions. Proposition 1 uses (A1)–(A6); (A7) gives the additional inclusion of the coupled class in the joint class. Proposition 2 uses (A1)–(A4), but does not require (A5)–(A7). Its risk-gap identity compares identical local heads; different heads generally change both $\bar{\varepsilon}$ and $\mathcal{O}$. The general identity equation 9 requires only (A1) and branch-restricted predictions. It covers the repeated-label setting, where a source with $\Delta_{\mathrm{TF}}(t,k) > \gamma_{\mathrm{TF}}$ can form a separate branch with an overlapping label set. In that setting, (A2) may fail and the unique-ownership comparisons are not invoked. The two main disjoint-label benchmarks satisfy (A2) by construction.

Complexity and estimation. For discrete predictors define the disagreement pseudometric $d_P(h,h') = \Pr_X[h(X) \neq h'(X)]$. Let $\mathrm{Comp}(\mathcal{H}) = \log \mathcal{N}(\epsilon_0, \mathcal{H}, d_P)$ be its covering entropy at a fixed scale $\epsilon_0 > 0$, using centers in the class. The same definition applies to routers. Under (A2) and fixed heads, $f_{q(x)}(x) \neq f_{q'(x)}(x)$ if and only if $q(x) \neq q'(x)$; thus the router class and its induced prediction class are isometric and $\mathrm{Comp}(\mathcal{H}_{\mathrm{dec}}) = \mathrm{Comp}(\mathcal{R})$.

For a statistical comparison, let $\mathcal{L}_{\mathcal{H}} = \{(x,y) \mapsto \mathbf{1}\{h(x) \neq y\} : h \in \mathcal{H}\}$ and let $\mathfrak{R}_n(\mathcal{L}_{\mathcal{H}})$ denote its expected Rademacher complexity on n i.i.d. observations. For a measurable class fixed independently of that sample, an empirical risk minimizer satisfies, with probability at least $1-\delta$,

$$R(\hat{h}_{\mathcal{H}}) - R^* \leq R(\mathcal{H}) - R^* + 4\mathfrak{R}_n(\mathcal{L}_{\mathcal{H}}) + 2\sqrt{\frac{2\log(4/\delta)}{n}}. \tag{8}$$

This follows by applying the bounded-loss uniform-convergence bound to both signs of the loss and comparing empirical risks (Shalev-Shwartz & Ben-David, 2014, Chapter 26). A term of order $\sqrt{p/n}$ requires additional control of the loss-class complexity; a covering number at one fixed scale alone does not supply that control. In particular, equation 8 is an i.i.d. reference comparison, not a generalization guarantee for the replay-free sequential optimizer.

For finite-dimensional parameterizations, adjustable parameter budgets provide a practical proxy for model complexity:

$$p_{\mathrm{cpl}} = 2K, \qquad p_{\mathrm{dec}} = p_{\mathcal{R}}, \qquad p_{\mathrm{joint}} = (d+1)|\mathcal{C}_{1:t}| + p_{\Phi}.$$

Here $p_{\mathcal{R}}$ includes all adjustable routing parameters, including source-fVAE parameters when they are fitted, and p_{Φ} counts the feature-map parameters. A larger class vocabulary increases the joint-head parameter budget for fixed d, whereas a repeated source need not add labels. Source-fVAE storage and scoring grow with the number of sources even when the number of task branches stays fixed.

Proof of Proposition 1. For each coupled predictor, κ_{cal} is a measurable function of $\boldsymbol{\xi}(x)$. A correct class prediction requires an admissible branch, so $R(\mathcal{H}_{\mathrm{cpl}}) \geq e^{\mathrm{cal}}_{\min}$. For any $q \in \mathcal{R}$, (A1)–(A4) give

$$R(f_q) = \bar{\varepsilon} + \Pr[q(x) \neq k^*,\, \mathcal{O}].$$

Taking the infimum over q yields $R(\mathcal{H}_{\mathrm{dec}}) = \bar{\varepsilon} + \eta_{\mathcal{R}}$. Condition (A5) then proves the first line of equation 5 by subtracting R^*. The isometry established above and (A6) give its second line; parameter counting is not used to prove that ordering.

Under (A7), each class has a unique owning branch by (A2). For $c \in \mathcal{Y}_k$, set $\widetilde{w}_c = a_k w_c$ and $\widetilde{\beta}_c = a_k \beta_c + b_k$. Then $T_k(s^{(k)}_c(x)) = \langle \widetilde{w}_c, \phi_0(x) \rangle + \widetilde{\beta}_c$. These scores and the consistent tie rule represent the same coupled predictor in $\mathcal{H}_{\mathrm{joint}}$. Hence $\mathcal{H}_{\mathrm{cpl}} \subseteq \mathcal{H}_{\mathrm{joint}}$ and $R(\mathcal{H}_{\mathrm{joint}}) \leq R(\mathcal{H}_{\mathrm{cpl}})$. □

B.1 Task-Identity Error and Classification Risk

For any predictor h whose output lies in $\bigcup_k \mathcal{Y}_k$, let $\kappa_{\mathrm{lab}}(h(x))$ select a branch owning its predicted label, using a fixed rule when that label belongs to several branches. Under (A1), $h(x) = y$ implies $\kappa_{\mathrm{lab}}(h(x)) \in \mathcal{K}^*(x)$, and therefore

$$\Pr[h(x) \neq y] \geq \Pr[\kappa_{\mathrm{lab}}(h(x)) \notin \mathcal{K}^*(x)].$$

With disjoint label sets, correct global classification necessarily identifies the unique branch owning the true label, including for a global-softmax model. With overlapping label sets, only label admissibility is necessary: selecting a branch containing the true label need not recover the originating source or its assigned branch. The routing term e_{route} below measures failure of this label-admissibility condition.

For the decoupled predictor, define

$$\mathcal{A}_{\mathrm{ok}} = \{\hat{k} \in \mathcal{K}^*(x)\}, \qquad A = \{f_{\hat{k}}(x) = y\}.$$

On $\mathcal{A}_{\mathrm{ok}}^c$, the true label is absent from $\mathcal{Y}_{\hat{k}}$, whereas $\hat{y} = f_{\hat{k}}(x) \in \mathcal{Y}_{\hat{k}}$. Thus $A \subseteq \mathcal{A}_{\mathrm{ok}}$, and the disjoint events $\mathcal{A}_{\mathrm{ok}}^c$ and $\mathcal{A}_{\mathrm{ok}} \cap A^c$ partition the classification-error event. Finite additivity gives the exact decomposition

$$\Pr[\hat{y} \neq y] = \Pr[\mathcal{A}_{\mathrm{ok}}^c] + \Pr[\mathcal{A}_{\mathrm{ok}} \cap A^c] = e_{\mathrm{route}} + \varepsilon_{\mathrm{local}}. \tag{9}$$

This identity requires (A1) and branch-local prediction, but not disjoint label sets. Under unique label ownership, it also yields an exact comparison with the same local classifiers supplied with the task identity.

Proposition 3 (Excess risk over a task-aware oracle)**.** *Assume (A1)–(A2), so each example has a unique admissible branch k^*. Let the task-aware oracle predict $f_{k^*}(x)$ using the same local classifiers. Then*

$$\Pr[\hat{y} \neq y] - \bar{\varepsilon} = \Pr[\hat{k} \neq k^*,\ f_{k^*}(x) = y] = e_{\mathrm{route}} - \Pr[\hat{k} \neq k^*,\ f_{k^*}(x) \neq y].$$

The first equality is the decoupled identity in equation 6. The second subtracts examples on which routing fails and the oracle also makes a local classification error, avoiding double counting.

Proof of Proposition 3 and the first identity in Proposition 2. Under (A2), $\mathcal{K}^*(x) = \{k^*\}$ and $\mathcal{A}_{\mathrm{ok}} = \{\hat{k} = k^*\}$. Write $\mathcal{O} = \{f_{k^*}(x) = y\}$. On $\mathcal{A}_{\mathrm{ok}}$, the routed and oracle predictions agree; on $\mathcal{A}_{\mathrm{ok}}^c$, the routed prediction is incorrect because its label set excludes y. Consequently, $\{\hat{y} = y\} = \mathcal{A}_{\mathrm{ok}} \cap \mathcal{O}$, giving

$$\begin{aligned}\Pr[\hat{y} \neq y] &= \Pr[\mathcal{O}^c] + \Pr[\mathcal{A}_{\mathrm{ok}}^c \cap \mathcal{O}] \\ &= \bar{\varepsilon} + \Pr[\hat{k} \neq k^*,\ f_{k^*}(x) = y].\end{aligned}$$

This proves the first equality of the proposition and the decoupled identity in equation 6. Partitioning the routing-error event gives

$$e_{\mathrm{route}} = \Pr[\mathcal{A}_{\mathrm{ok}}^c] = \Pr[\mathcal{A}_{\mathrm{ok}}^c \cap \mathcal{O}] + \Pr[\mathcal{A}_{\mathrm{ok}}^c \cap \mathcal{O}^c],$$

which proves the second equality. In particular, the excess risk lies in $[0, e_{\mathrm{route}}]$, and $\Pr[\hat{y} \neq y] \leq \bar{\varepsilon} + e_{\mathrm{route}}$. The nonnegativity of $\varepsilon_{\mathrm{local}}$ in equation 9 also gives $\Pr[\hat{y} \neq y] \geq e_{\mathrm{route}}$. □

The overlap $\mathcal{A}_{\mathrm{ok}}^c \cap \mathcal{O}^c$ contains inputs assigned to a wrong branch that their own branch would also misclassify. These examples already contribute to $\bar{\varepsilon}$; charging them again would overstate the additional classification error due to routing. This oracle comparison uses unique label ownership, whereas equation 9 remains valid when several branches contain the true label.

Corollary 1 (Class-balanced form)**.** *Assume (A1) and* $\Pr[y = c] > 0$ *for each* $c \in \mathcal{C}_{1:t}$*. Let* $\mathrm{mAcc} = |\mathcal{C}_{1:t}|^{-1} \sum_{c \in \mathcal{C}_{1:t}} \Pr[\hat{y} = y \mid y = c]$ *be the unweighted average class recall defined in Appendix F. Define the class-balanced terms*

$$\bar{e}_{\mathrm{route}} = \frac{1}{|\mathcal{C}_{1:t}|} \sum_{c \in \mathcal{C}_{1:t}} \Pr[\hat{k} \notin \mathcal{K}^*(x) \mid y = c], \qquad \bar{\varepsilon}_{\mathrm{local}} = \frac{1}{|\mathcal{C}_{1:t}|} \sum_{c \in \mathcal{C}_{1:t}} \Pr[\hat{k} \in \mathcal{K}^*(x),\, f_{\hat{k}}(x) \neq y \mid y = c].$$

Then $1 - \mathrm{mAcc} = \bar{e}_{\mathrm{route}} + \bar{\varepsilon}_{\mathrm{local}}$*. The barred quantities weight classes equally, whereas* e_{route} *and* $\varepsilon_{\mathrm{local}}$ *use the test distribution's class frequencies.*

Proof of Corollary 1. The disjoint-event decomposition in equation 9 remains valid after conditioning on $y = c$:

$$\Pr[\hat{y} \neq y \mid y = c] = \Pr[\hat{k} \notin \mathcal{K}^*(x) \mid y = c] + \Pr[\hat{k} \in \mathcal{K}^*(x),\, f_{\hat{k}}(x) \neq y \mid y = c].$$

Averaging over classes with weights $1/|\mathcal{C}_{1:t}|$ gives $1 - \mathrm{mAcc}$ on the left and the two class-balanced error terms on the right. This identity applies to the mAcc results in section 5. It does not provide the same additive decomposition for Macro-F1, which also depends on class precision. □

Corollary 2 (Source-to-task aggregation)**.** *After learning* t *sources, let* $k(s)$ *be the branch assigned to source* $s \leq t$*, with* $\mathcal{Y}_{k(s)} = \mathcal{C}_s$*, and let* s^* *be the true source of a test example, so* $y \in \mathcal{C}_{s^*}$*. Choose*

$$\hat{s} \in \arg\max_{s \leq t} \psi_s^{\mathrm{recon}}(G(x))$$

with fixed tie-breaking, and resolve ties in the branch-level maximization consistently so that $\hat{k} = k(\hat{s})$*, as permitted by the max aggregation in Algorithm 1. Then*

$$e_{\mathrm{route}} \leq \Pr[k(\hat{s}) \neq k(s^*)] \leq \Pr[\hat{s} \neq s^*].$$

The branch-identity error satisfies

$$\begin{aligned} \Pr[k(\hat{s}) \neq k(s^*)] = \Pr[\hat{s} \neq s^*] \\ - \Pr[\hat{s} \neq s^*,\, k(\hat{s}) = k(s^*)]. \end{aligned}$$

In particular, $e_{\mathrm{route}} < \Pr[\hat{s} \neq s^*]$ *whenever within-branch source confusion has positive probability.*

Proof of Corollary 2. The eligibility rule in subsection 3.2 permits a merge only when $\mathcal{C}_s = \mathcal{Y}_{k(s)}$; a new branch is initialized with the same label set. Hence the true source's branch contains y. If $\hat{s} = s^*$, then $k(\hat{s}) = k(s^*)$. If $k(\hat{s}) = k(s^*)$, then $\hat{k} \in \mathcal{K}^*(x)$. Taking complements gives the event inclusions

$$\{\hat{k} \notin \mathcal{K}^*(x)\} \subseteq \{k(\hat{s}) \neq k(s^*)\} \subseteq \{\hat{s} \neq s^*\},$$

and thus the claimed inequalities. Partitioning $\{\hat{s} \neq s^*\}$ according to whether the two sources share a branch gives the exact branch-identity expression. If its subtracted event has positive probability, the branch-identity error, and therefore also e_{route}, is strictly smaller than the source-identity error. □

This result distinguishes branch identity from label admissibility. Grouping sources removes the need to distinguish their identities within a shared branch. However, distinct branches with identical label sets are already interchangeable for the label-admissibility criterion, though their local predictions may differ. In particular, if source scores and the selected maximizing source $\hat{s}$ are held fixed, grouping only identical-label sources leaves

$$e_{\mathrm{route}} = \Pr[y \notin \mathcal{Y}_{k(\hat{s})}] = \Pr[y \notin \mathcal{C}_{\hat{s}}]$$

unchanged. The corollary therefore bounds branch and label-admissibility errors by source-identification error; it does not establish that merging improves total classification risk.

Branch isolation is intended to preserve task-local classification: unmatched task branches remain frozen, whereas a matched branch can be reopened under Task Formation. Under unique label ownership, its local quality contributes to $\bar{\varepsilon}$; in the general decomposition, local errors are represented by $\varepsilon_{\text{local}}$. The KD term in equation 4 anchors a reused branch to its pre-update outputs on current-source inputs only, without guaranteeing unchanged risk on earlier sources. The router determines the label-admissibility term in equation 9. Empirically, reconstruction routing brings final performance close to the task-identity oracle in Figure 4.

B.2 Limits of Cross-Source Score Calibration

This subsection characterizes the limits of task selection based on recalibrated local class scores. We first prove the calibrated-predictor identity and risk comparison in Proposition 2, then examine a stylized additive-offset model.

Proof of the calibrated-risk identity and risk comparison in Proposition 2. Because T_k is strictly increasing, it preserves the ranking within branch k:

$$\max_{c\in\mathcal{Y}_k} T_k\Big(s_c^{(k)}(x)\Big) = T_k\left(\max_{c\in\mathcal{Y}_k} s_c^{(k)}(x)\right) = T_k\big(\xi_k(x)\big).$$

The calibrated global maximum is taken over pairs (k, c) with $c \in \mathcal{Y}_k$. Use the same fixed branch-first tie-breaking rule for this maximum and for κ_{cal}, followed by the fixed within-branch rule used by f_k. Thus the selected label is $\hat{y}_{\text{cal}} = f_{\kappa_{\text{cal}}(x)}(x)$, where $\kappa_{\text{cal}}(x) = \arg\max_k T_k(\xi_k(x))$ depends on x only through $\boldsymbol{\xi}(x)$. This conclusion also holds when all relevant maxima are unique almost surely.

On $\{\kappa_{\text{cal}} = k^*\}$, the calibrated rule uses f_{k^*} and is correct exactly when $\mathcal{O}$ holds. On $\{\kappa_{\text{cal}} \neq k^*\}$, disjoint task label sets imply $y \notin \mathcal{Y}_{\kappa_{\text{cal}}}$, so the prediction is incorrect. Consequently, $\{\hat{y}_{\text{cal}} = y\} = \{\kappa_{\text{cal}} = k^*\} \cap \mathcal{O}$ and

$$\Pr[\hat{y}_{\text{cal}} \neq y] = \bar{\varepsilon} + \Pr[\kappa_{\text{cal}} \neq k^*,\, \mathcal{O}].$$

A correct class prediction also implies $\kappa_{\text{cal}}(x) \in \mathcal{K}^*(x)$. Since κ_{cal} is a measurable function of $\boldsymbol{\xi}$, it follows that $\Pr[\hat{y}_{\text{cal}} \neq y] \geq \Pr[\kappa_{\text{cal}} \notin \mathcal{K}^*] \geq e_{\min}^{\text{cal}}$. For the same fixed heads, subtracting the two identities in equation 6 cancels $\bar{\varepsilon}$. The event on which both routers select an incorrect branch and $\mathcal{O}$ holds also cancels, leaving exactly the two disagreement events in equation 7. □

The lower bound concerns the information retained by $\boldsymbol{\xi}(x)$. Additional calibration data can improve a fitted rule, but no rule based only on this statistic can achieve routing error below $e_{\min}^{\text{cal}}$. Under disjoint task label sets, let $p_k(\boldsymbol{\xi}) = \Pr[k^* = k \mid \boldsymbol{\xi}(x) = \boldsymbol{\xi}]$. Then

$$e_{\min}^{\text{cal}} = 1 - \mathbb{E}\left[\max_{k\leq K} p_k(\boldsymbol{\xi}(x))\right],$$

so the lower bound is positive precisely when $\Pr[\max_k p_k(\boldsymbol{\xi}(x)) < 1] > 0$. This condition expresses residual task ambiguity given the score statistic; overlap of the topological supports alone is insufficient. A local cosine head can assign a high top score to an input from another source because its local objective does not include other-source negatives. This makes score ambiguity possible, but does not by itself establish a positive lower bound.

The source-fVAE router evaluates $r = G(x)$ and is not restricted to functions of $\boldsymbol{\xi}(x)$, so this statistic-specific lower bound does not automatically apply to it. Its advantage nevertheless depends on the informativeness of its reconstruction evidence. In Table 2, fitted affine calibration remains 5.94 Macro-F1 points below task-first inference. This observed metric gap supports the empirical comparison, but neither estimates $e_{\min}^{\text{cal}}$ nor proves that it is positive. The theoretical distinction is that per-branch monotone calibration selects a task from the K branch-wise maxima, whereas a separate router can use additional evidence about the appropriate task branch.

A stylized instance: additive offsets. To isolate one failure mode of direct global prediction, consider additive branch offsets under the disjoint-label condition (A2). Each class c then has a unique owning branch $\kappa_{\mathrm{lab}}(c)$. Let

$$\tilde{s}_c(x) = s_c(x) + b_{\kappa_{\mathrm{lab}}(c)},$$

where $s_c(x)$ determines the within-branch ranking and b_k is a branch-dependent score offset. The global rule predicts

$$\hat{y}_G = \arg \max_{c \in \bigcup_k \mathcal{Y}_k} \tilde{s}_c(x).$$

For an input from task branch k^* whose within-branch classifier is correct,

$$s_y(x) = \max_{c \in \mathcal{Y}_{k^*}} s_c(x),$$

define its margin against branch k' as

$$M_{k^* \to k'}(x) = s_y(x) - \max_{c \in \mathcal{Y}_{k'}} s_c(x).$$

Proposition 4 (Bias-induced wrong-task prediction)**.** *Assume (A2) and that the shifted branch maxima $b_k + \max_{c \in \mathcal{Y}_k} s_c(x)$ have a unique maximizer almost surely. For an input whose true-branch classifier is correct, the global prediction $\hat{y}_G$ leaves the true task branch if and only if*

$$\exists\, k' \neq k^* : \qquad b_{k'} - b_{k^*} > M_{k^* \to k'}(x).$$

Proof of Proposition 4. Local correctness gives $s_y(x) = \max_{c \in \mathcal{Y}_{k^*}} s_c(x)$. Under the stated uniqueness condition, the global prediction leaves the true task branch exactly when an off-task branch maximum strictly exceeds the true-branch maximum:

$$b_{k'} + \max_{c \in \mathcal{Y}_{k'}} s_c(x) > b_{k^*} + s_y(x)$$

for some $k' \neq k^*$. Rearranging yields

$$b_{k'} - b_{k^*} > s_y(x) - \max_{c \in \mathcal{Y}_{k'}} s_c(x) = M_{k^* \to k'}(x),$$

which proves the equivalence. If shifted branch maxima can tie, the strict inequality still suffices; the equality case depends on the fixed tie-breaking rule and must be included for a necessary-and-sufficient characterization. □

Thus a correct within-task decision need not imply a correct global decision. Branch-dependent offsets can make an off-task score exceed the true-branch maximum, as characterized by Proposition 4.

Corollary 3 (Effect of offset equalization)**.** *If $b_k \equiv b$ for all branches, the offsets preserve the global ranking of the base scores. Conversely, unequal offsets can turn a correct base-score global prediction into a wrong-task prediction when an offset difference exceeds a positive cross-branch margin. Equalizing the offsets alone does not guarantee that the base-score global prediction is correct.*

Proof. If all offsets are equal, fixed tie-breaking gives

$$\arg\max_c \tilde{s}_c(x) = \arg\max_c (s_c(x) + b) = \arg\max_c s_c(x).$$

Conversely, suppose $b_{k'} > b_k$. Consider a locally correct input from branch k for which $M_{k \to j}(x) > 0$ for every $j \neq k$, so the base-score global rule is correct. If, in addition,

$$0 < M_{k \to k'}(x) < b_{k'} - b_k,$$

then the off-task shifted maximum strictly exceeds the true-branch maximum. The condition of Proposition 4 therefore holds, and the shifted global rule leaves the true task branch. If the base scores already misidentify the task, equalizing the offsets preserves that error. □

The additive-offset model is deliberately limited: equalization removes the branch-dependent constants, but leaves any errors in the base-score ranking. If the base-score maxima already identify the true task almost surely, the correction $T_k(u) = u - b_k$ yields perfect task selection from the shifted scores, and the corresponding $e_{\min}^{\mathrm{cal}}$ is zero. Without this additional condition, offset equalization does not imply a zero routing lower bound.

The 5.94-point Macro-F1 gap after fitted task-wise affine calibration shows that this fitted rule does not match task-first inference under the reported protocol. It does not establish that all remaining mismatch is input-dependent, rule out a better-fitted affine correction, or determine the optimal performance of arbitrary monotone corrections. Proposition 2 instead gives a conditional limitation: if task ambiguity remains in $\boldsymbol{\xi}(x)$, no per-branch monotone transformation can remove all routing errors using that statistic alone. Once HiDMoA selects a task branch, class-score distortions across branches no longer enter its final within-task $\arg\max$; errors in the preceding routing decision can still affect the final prediction.

B.3 Routing under Separated Task Evidence

We analyze routing at a fixed stage after observing t sources, conditioning on the learned source fVAEs and a fixed test feature $r = G(x) \in \mathbb{R}^m$. All t source fVAEs use the same frozen routing encoder G. For source $s \le t$, write

$$X_s(z; r) = \frac{1}{m}\|r - \hat{r}_s(z)\|_2^2, \qquad z \sim q_s(z \mid r),$$

and distinguish the posterior-expected reconstruction score from its Monte Carlo estimate:

$$\Psi_s^{\mathrm{recon}}(r) = -\mathbb{E}_{q_s(z|r)}[X_s(z; r)],$$
$$\widehat{\psi}_{s,N_{\mathrm{mc}}}^{\mathrm{recon}}(r) = -\frac{1}{N_{\mathrm{mc}}}\sum_{j=1}^{N_{\mathrm{mc}}} X_s(z_s^{(j)}; r), \qquad z_s^{(j)} \sim q_s(z \mid r).$$

Thus $\widehat{\psi}_{s,N_{\mathrm{mc}}}^{\mathrm{recon}}$ is precisely the finite-sample score denoted by ψ_s^{recon} in equation 2; Ψ_s^{recon} denotes its expectation only in this analysis. In the basic one-source/one-task setting, source and task indices coincide.

Reconstruction evidence and marginal likelihood. The common Gaussian decoder model

$$p_s(r \mid z) = \mathcal{N}(\hat{r}_s(z), I)$$

gives

$$\log p_s(r \mid z) = -\frac{1}{2}\|r - \hat{r}_s(z)\|_2^2 - \frac{m}{2}\log(2\pi).$$

The frozen feature coordinates, dimensional normalization, and decoder scale therefore provide a common measurement basis. They do not guarantee cross-source calibration: source distributions and reconstruction difficulty may still differ.

Assuming the expectations and divergences below are finite, define

$$\Gamma_s(r) = \mathrm{KL}(q_s(z \mid r)\|p(z)) - \mathrm{KL}(q_s(z \mid r)\|p_s(z \mid r)).$$

The standard evidence decomposition and $\mathbb{E}_{q_s}[\log p_s(r \mid z)] = \frac{m}{2}\Psi_s^{\mathrm{recon}}(r) - \frac{m}{2}\log(2\pi)$ imply the exact identity

$$\log p_s(r) = \frac{m}{2}\Psi_s^{\mathrm{recon}}(r) - \frac{m}{2}\log(2\pi) - \Gamma_s(r). \tag{10}$$

Thus differences between expected reconstruction and marginal-likelihood comparisons are determined by the variation of $\Gamma_s(r)$ across sources. The identity concerns expected scores; finite-sample scores incur an additional Monte Carlo error, analyzed below.

Lemma 1 (Agreement of reconstruction and likelihood maximizers)**.** *Suppose at least two source fVAEs are stored. Define* $\mathrm{osc}_\Gamma(r) = \max_{s\le t}\Gamma_s(r) - \min_{s\le t}\Gamma_s(r)$*, and let* $\tilde\gamma_{\mathrm{recon}}(r)$ *be the gap between the largest and second-largest expected reconstruction scores* $\Psi_s^{\mathrm{recon}}(r)$*. If*

$$\tilde\gamma_{\mathrm{recon}}(r) > \frac{2}{m}\mathrm{osc}_\Gamma(r),$$

then the two criteria have the same unique maximizer, $\arg\max_{s\le t}\Psi_s^{\mathrm{recon}}(r) = \arg\max_{s\le t}\log p_s(r)$.

Proof. Let s_0 maximize $\Psi_s^{\mathrm{recon}}(r)$. By equation 10, for every $s \neq s_0$,

$$\log p_{s_0}(r) - \log p_s(r) = \frac{m}{2}\big(\Psi_{s_0}^{\mathrm{recon}}(r) - \Psi_s^{\mathrm{recon}}(r)\big) - \big(\Gamma_{s_0}(r) - \Gamma_s(r)\big) \ge \frac{m}{2}\tilde\gamma_{\mathrm{recon}}(r) - \mathrm{osc}_\Gamma(r) > 0.$$

The strict gap also makes the reconstruction maximizer unique. □

This lemma gives a sufficient condition for agreement of the maximizing source, not the complete source ordering or recovery of the generating source. It does not establish that reconstruction always outperforms a likelihood-based or KL-augmented score. In particular, let $D_s^{\mathrm{prior}} = \mathrm{KL}(q_s(z \mid r)\|p(z))$ and $D_s^{\mathrm{post}} = \mathrm{KL}(q_s(z \mid r)\|p_s(z \mid r))$. The expectation of the weighted Recon+KL score in equation 11 is

$$\begin{aligned}\Psi_s^{\mathrm{Recon+KL}}(r) &= \lambda_{\mathrm{rec}}\Psi_s^{\mathrm{recon}}(r) - \beta D_s^{\mathrm{prior}} \\ &= \frac{2\lambda_{\mathrm{rec}}}{m}\log p_s(r) + \lambda_{\mathrm{rec}}\log(2\pi) + \left(\frac{2\lambda_{\mathrm{rec}}}{m} - \beta\right) D_s^{\mathrm{prior}} - \frac{2\lambda_{\mathrm{rec}}}{m} D_s^{\mathrm{post}}.\end{aligned}$$

Only when $\beta = 2\lambda_{\mathrm{rec}}/m$ does the prior-divergence term cancel in this expression; the source-dependent posterior gap remains. Its variation may improve or worsen the comparison relative to reconstruction alone.

The comparison between Recon and Recon+KL in Figure 4 is empirical: their task-identity accuracies are nearly identical at the displayed precision, and adding KL shows no clear benefit in that comparison. The lemma establishes no universal ordering between these scores. Using KL during training to regularize each source posterior toward the common prior is a separate role.

Separation and Monte Carlo error. In the nontrivial one-source/one-task setting with $t \ge 2$, let s^* be the source generating the test input and define its expected reconstruction margin and the maximum estimation error by

$$\gamma_{\mathrm{recon}}(r) = \Psi_{s^*}^{\mathrm{recon}}(r) - \max_{s\neq s^*}\Psi_s^{\mathrm{recon}}(r),$$

$$\delta_{N_{\mathrm{mc}}}(r) = \max_{s\le t}\left|\widehat\psi_{s,N_{\mathrm{mc}}}^{\mathrm{recon}}(r) - \Psi_s^{\mathrm{recon}}(r)\right|.$$

Unlike the top-two gap in Lemma 1, $\gamma_{\mathrm{recon}}(r)$ need not be positive: the generating source need not have the highest expected reconstruction score.

Proposition 5 (Correct routing under separated reconstruction scores)**.** *If*

$$\gamma_{\mathrm{recon}}(r) > 2\delta_{N_{\mathrm{mc}}}(r),$$

then

$$\arg\max_{s\le t}\widehat\psi_{s,N_{\mathrm{mc}}}^{\mathrm{recon}}(r) = s^*.$$

Proof. For the generating source, $\widehat\psi_{s^*,N_{\mathrm{mc}}}^{\mathrm{recon}}(r) \ge \Psi_{s^*}^{\mathrm{recon}}(r) - \delta_{N_{\mathrm{mc}}}(r)$, whereas every $s \neq s^*$ satisfies $\widehat\psi_{s,N_{\mathrm{mc}}}^{\mathrm{recon}}(r) \le \Psi_s^{\mathrm{recon}}(r) + \delta_{N_{\mathrm{mc}}}(r)$. Therefore,

$$\begin{aligned}\widehat\psi_{s^*,N_{\mathrm{mc}}}^{\mathrm{recon}}(r) - \widehat\psi_{s,N_{\mathrm{mc}}}^{\mathrm{recon}}(r) &\ge \Psi_{s^*}^{\mathrm{recon}}(r) - \Psi_s^{\mathrm{recon}}(r) - 2\delta_{N_{\mathrm{mc}}}(r) \\ &\ge \gamma_{\mathrm{recon}}(r) - 2\delta_{N_{\mathrm{mc}}}(r) > 0.\end{aligned}$$

Thus s^* has the largest estimated reconstruction score. □

Under Task Formation, define the population and Monte Carlo task-branch scores by aggregating the corresponding source scores:

$$\Psi_k^{\mathrm{task}}(r) = \max_{s \in \mathcal{S}_k} \Psi_s^{\mathrm{recon}}(r), \qquad \widehat{\psi}_{k,N_{\mathrm{mc}}}^{\mathrm{task}}(r) = \max_{s \in \mathcal{S}_k} \widehat{\psi}_{s,N_{\mathrm{mc}}}^{\mathrm{recon}}(r).$$

The second expression is the task-branch score used at inference. Since $|\max_s a_s - \max_s b_s| \leq \max_s |a_s - b_s|$, max aggregation does not amplify the worst source-score estimation error:

$$\max_{k \leq K_t} \left| \widehat{\psi}_{k,N_{\mathrm{mc}}}^{\mathrm{task}}(r) - \Psi_k^{\mathrm{task}}(r) \right| \leq \delta_{N_{\mathrm{mc}}}(r).$$

The same proof gives recovery of a specified task branch whenever its population score exceeds every competing branch score by more than $2\delta_{N_{\mathrm{mc}}}(r)$. For the label-admissible routing criterion in the risk decomposition, let $\mathcal{K}^* = \{k \leq K_t : y \in \mathcal{Y}_k\}$ for the test pair (x, y). When there is at least one inadmissible branch, define

$$\gamma_{\mathrm{task}}(r) = \max_{k \in \mathcal{K}^*} \Psi_k^{\mathrm{task}}(r) - \max_{k \notin \mathcal{K}^*} \Psi_k^{\mathrm{task}}(r).$$

If $\gamma_{\mathrm{task}}(r) > 2\delta_{N_{\mathrm{mc}}}(r)$, a maximizing admissible branch retains a larger estimated score than every inadmissible branch, so $\hat{k} \in \mathcal{K}^*$. If all branches are admissible, label-admissible routing is automatic. Selecting an admissible branch still leaves the within-branch classification error analyzed in the risk decomposition.

Proposition 5 distinguishes two requirements. A shared feature-space metric alone does not guarantee correct routing: the expected scores must favor the correct source or an admissible branch, and their margin must dominate the Monte Carlo error. A small or negative margin captures ambiguous or misleading reconstruction evidence, whereas $\delta_{N_{\mathrm{mc}}}(r)$ concerns only approximation of the expected scores. The high task-recovery accuracy in Figure 4 is empirical support for the routing mechanism on the evaluated streams, rather than verification of these pointwise conditions.

A conditional sampling-budget bound. The Monte Carlo error admits an explicit bound under an additional boundedness assumption.

Corollary 4 (Sampling budget for correct routing)**.** *Fix the learned models and r. Suppose $X_s(z; r) \in [0, \Omega_{\mathrm{rec}}]$ almost surely under $q_s(z \mid r)$ for all $s \leq t$, where $\Omega_{\mathrm{rec}} > 0$, and draw N_{mc} independent posterior samples within each source. For $\eta \in (0, 1)$, with probability at least $1 - \eta$ over these samples,*

$$\delta_{N_{\mathrm{mc}}}(r) \leq \Omega_{\mathrm{rec}} \sqrt{\frac{\log(2t/\eta)}{2N_{\mathrm{mc}}}}.$$

If additionally $\gamma_{\mathrm{recon}}(r) > 0$, then routing to the generating source in the one-source/one-task setting is correct with probability at least $1 - \eta$ whenever

$$N_{\mathrm{mc}} > \frac{2\Omega_{\mathrm{rec}}^2 \log(2t/\eta)}{\gamma_{\mathrm{recon}}(r)^2}.$$

For Task Formation, the same bound guarantees label-admissible routing when $\gamma_{\mathrm{recon}}(r)$ is replaced by a positive $\gamma_{\mathrm{task}}(r)$.

Proof. For any $u > 0$, Hoeffding's inequality for each source and a union bound give

$$\Pr[\delta_{N_{\mathrm{mc}}}(r) > u] \leq 2t \exp\left(-\frac{2N_{\mathrm{mc}} u^2}{\Omega_{\mathrm{rec}}^2} \right).$$

Taking $u = \Omega_{\mathrm{rec}} \sqrt{\log(2t/\eta)/(2N_{\mathrm{mc}})}$ proves the error bound. The sample-size condition makes $2u < \gamma_{\mathrm{recon}}(r)$, so Proposition 5 applies on this event. The task-branch conclusion follows from the max-aggregation error bound and the admissible-set margin. □

The union bound counts the t scored source fVAEs, which may exceed the K_t task branches. This is a pointwise guarantee for fixed learned models and a fixed test feature, not a uniform guarantee over test inputs. Moreover, a Gaussian posterior and the stated decoder architecture do not themselves ensure a finite almost-sure bound Ω_{rec}; the corollary therefore requires an additional assumption

and does not certify the default sampling budget. Under its assumptions, the sufficient budget is logarithmic in the number of scored source models and quadratic in the bound-to-margin ratio.

The sampling study in Table 6 complements this conditional analysis. At $N_{\mathrm{mc}} = 6$, Last Macro-F1 is 92.54 ± 0.66; increasing the budget to $N_{\mathrm{mc}} = 2000$ changes this metric by less than the reported seed standard deviations and mainly improves the Avg metrics. These results show limited sensitivity of the reported Last metrics to the tested budgets. They are consistent with stable reconstruction-based routing, but do not measure $\gamma_{\mathrm{recon}}(r)$ or $\delta_{N_{\mathrm{mc}}}(r)$ and therefore cannot establish which quantity limits routing accuracy. Freezing G preserves the feature coordinate system; it does not by itself guarantee a positive or unchanging reconstruction margin.

Incremental router cost. Each new source fVAE is trained only on frozen features from the current source and does not require revisiting real samples from previous sources. The current implementation stores one source fVAE per source, while task branches grow with the number of formed tasks. With fixed fVAE size and N_{mc}, router storage and per-input source scoring grow linearly with the number of observed sources.

C Algorithmic Summary

This section spells out the training and inference steps used in the implementation. Algorithm 1 first determines whether the arriving source forms a new task branch or joins an existing one, then follows the corresponding new-task-branch or merge update, registers the source fVAE, and uses task-branch-aggregated source-fVAE evidence at inference.

Algorithm 1: HiDMoA training and inference

Input: incremental stream $\{(\mathcal{D}_t^{\mathrm{train}}, \mathcal{D}_t^{\mathrm{val}}, \mathcal{C}_t)\}_{t=1}^{T}$, frozen classification backbone F, frozen routing encoder G, global threshold γ_{TF}, and a test input x at inference.

Output: task-branch modules $\{A^k, H_k, \mathcal{Y}_k, \mathcal{S}_k\}_{k=1}^{K_T}$, source fVAEs $\{V_t\}_{t=1}^{T}$, and prediction $\hat{y}$. Here $A^k = \{A_\ell^k\}_{\ell \in \mathcal{L}}$.

1. **For** each source $t = 1, \ldots, T$, fit the source fVAE V_t on the current frozen routing features $\{G(x) : (x, y) \in \mathcal{D}_t^{\mathrm{train}}\}$ and form the eligible set $\mathcal{E}_t = \{k \le K_{t-1} : \mathcal{C}_t = \mathcal{Y}_k\}$.
2. If $\mathcal{E}_t \neq \varnothing$, compute $\Delta_{\mathrm{TF}}(t, k)$ on $\mathcal{D}_t^{\mathrm{val}}$ and set a_t by the assignment rule in Section 3.2; otherwise set $a_t = \mathrm{new}$.
3. **New task branch:** if $a_t = \mathrm{new}$, set $k = K_{t-1} + 1$, $\mathcal{Y}_k = \mathcal{C}_t$, initialize A^k from the most related previous task branch using $j^*(t)$ and $e^*(t)$ when $t > 1$, initialize H_k by class-mean imprinting on $\mathcal{D}_t^{\mathrm{train}}$, and train (A^k, H_k) with equation 3. Set $\mathcal{S}_k = \{t\}$ and $K_t = k$.
4. **Merge into task branch:** if $a_t = k \le K_{t-1}$, keep $\mathcal{Y}_k$ unchanged and allocate no new task-branch adapters or local cosine head. Freeze a pre-update teacher copy of (A^k, H_k), select λ_{kd} by the current-source probe, and update only (A^k, H_k) using $\mathcal{L}_{\mathrm{update}}$. Then set $\mathcal{S}_k \leftarrow \mathcal{S}_k \cup \{t\}$ and $K_t = K_{t-1}$.
5. Register V_t under the assigned task branch; no previous-source train or validation samples are retained for replay or later tuning. **End For.**
6. **At inference**, compute $r = G(x)$, evaluate every source fVAE, aggregate source-level reconstruction scores within each task branch as $\psi_k^{\mathrm{task}}(r) = \max_{s \in \mathcal{S}_k} \psi_s^{\mathrm{recon}}(r)$, select $\hat{k} = \arg\max_{k \le K_t} \psi_k^{\mathrm{task}}(r)$, activate $(A^{\hat{k}}, H_{\hat{k}})$, and return $\hat{y} = \arg\max_{c \in \mathcal{Y}_{\hat{k}}} s_c^{(\hat{k})}(x)$.

D Additional Ablations

These ablations use the same order-1 industrial split and seeds as the main ablation section, with DeiT-Small by default and an additional ResNet-18 comparison for adapter placement. They support the design choices used in the main experiments by isolating the routing score, expert initialization, adapter position, reconstruction quality, sampling budget, and top-k inference.

D.1 Routing Scores and Decoupled Inference

At inference, every seen source fVAE evaluates the same frozen routing feature $r = G(x)$. The default reconstruction score equation 2 is defined in subsection 3.2, while task-branch aggregation, task selection, and local prediction are defined in subsection 3.4. This subsection records the two alternative source-fVAE scores used in the routing ablation and the optional top-2 correction.

The default source-fVAE score is the negative reconstruction error already defined in equation 2: a larger $\psi_s^{\mathrm{recon}}(r)$ means that source fVAE V_s explains the frozen routing feature better. Reconstruction-only routing is the default because it uses the same frozen representation and metric as fVAE training; the KL term remains a training regularizer, while subsection B.3 studies the approximation error of reconstruction-based routing.

The first alternative follows the generative-classifier view and estimates the marginal likelihood by importance sampling (Van De Ven et al., 2021),

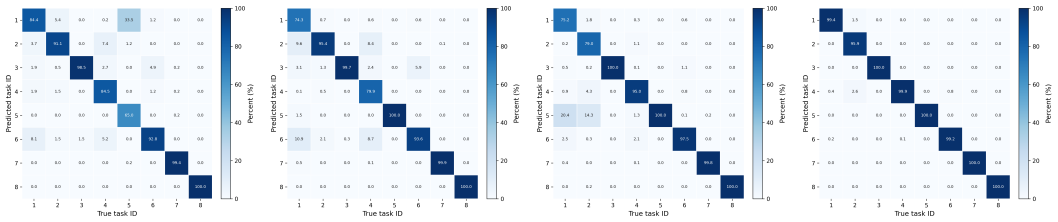
$$\psi_s^{\mathrm{IS}}(r) = \log \left[\frac{1}{N_{\mathrm{mc}}} \sum_{j=1}^{N_{\mathrm{mc}}} \frac{p_s(r|z^{(j)})p(z^{(j)})}{q_s(z^{(j)}|r)} \right], \qquad z^{(j)} \sim q_s(z|r),$$

with N_{mc} importance samples. The second uses the full negative fVAE training objective and therefore retains the same fixed λ_{rec} and β as equation 1,

$$\psi_s^{\mathrm{Recon+KL}}(r) = -\frac{1}{N_{\mathrm{mc}}} \sum_{j=1}^{N_{\mathrm{mc}}} \left[\lambda_{\mathrm{rec}} \frac{1}{m} \|r - \hat{r}_s(z^{(j)})\|_2^2 + \beta \, \mathrm{KL}(q_s(z|r)\|p(z)) \right], \qquad z^{(j)} \sim q_s(z|r). \tag{11}$$

Thus Recon, IS, and Recon+KL differ only in the evidence assigned by an individual source fVAE. For any one of these source-level scores ψ_s^q, the corresponding task-branch aggregate is $\psi_k^{\mathrm{task},q}(r) = \max_{s \in \mathcal{S}_k} \psi_s^q(r)$. The main method uses $q = \mathrm{recon}$ as in the inference step of Algorithm 1.

The heatmaps in Figure 6 show where the routing gains come from. They provide supplementary evidence for the task-identity inference ablation in Figure 4. Feature prototypes already separate the easiest sources, but they produce systematic leaks, sending many `dagm` samples to `neu` and mixing `cr7` with `xsdd` and `gc10`. Class-level fVAEs reduce some of these errors, while source-fVAE importance sampling keeps `dagm` clean but over-assigns other sources to it. Reconstruction scoring removes most of these systematic leaks; the remaining errors are small and local, mainly `xsdd` confused with neighboring sources.



Task Prototype | Class fVAE | Source fVAE (IS) | Source fVAE (Recon)

Figure 6: Source-confusion matrices after the final incremental step, averaged over three seeds. Columns indicate true sources and rows predicted sources. Each source corresponds to one task in this experiment.

Decoupled inference. After the router selects $\hat{k}$, inference activates only $(A^{\hat{k}}, H_{\hat{k}})$ and predicts within $\mathcal{Y}_{\hat{k}}$, as summarized in the inference step of Algorithm 1. Thus the alternative scores above change only task recovery; they do not introduce a global competition among independently trained class heads.

The top-1 rule is efficient but can be brittle for low-margin samples whose two best formed-task scores are close. We therefore retain an optional confidence-aware top-2 correction, which is not

used in the main setting unless explicitly stated. Let

$$\ell_k(r) = \psi_k^{\text{task,recon}}(r), \qquad \omega_k(r) = \text{softmax}(\ell(r))_k, \qquad k = 1, \ldots, K_t. \tag{12}$$

Let k_1, k_2 denote the task branches with the largest and second-largest normalized routing weights, respectively. If

$$\omega_{k_1}(r) - \omega_{k_2}(r) \geq \alpha / K_t, \tag{13}$$

the model keeps the top-1 branch. Otherwise it activates both candidates and returns the label with the largest joint task-class score,

$$J(k_i, c|x) = \log \omega_{k_i}(r) + s_c^{(k_i)}(x), \qquad i \in \{1, 2\}, \quad c \in \mathcal{Y}_{k_i}. \tag{14}$$

This remains distinct from full global classification: at most two local cosine heads are compared, and the routing weight remains part of the decision. The same correction can be applied to another task-branch score by replacing ℓ_k accordingly; the sensitivity experiment below uses task-branch-aggregated IS scores to match its original experimental configuration.

D.2 Expert Initialization and Similarity-Guided Old-Expert Reuse

New-Task Expert Initialization. A new source can initialize its adapter from random weights or by copying a related previous expert. In a multi-source stream, related sources can provide a better starting point than random initialization. We use Recon routing with $N_{\text{mc}} = 600$ and vary the number of new experts and whether their initial parameters are copied from the most related previous experts.

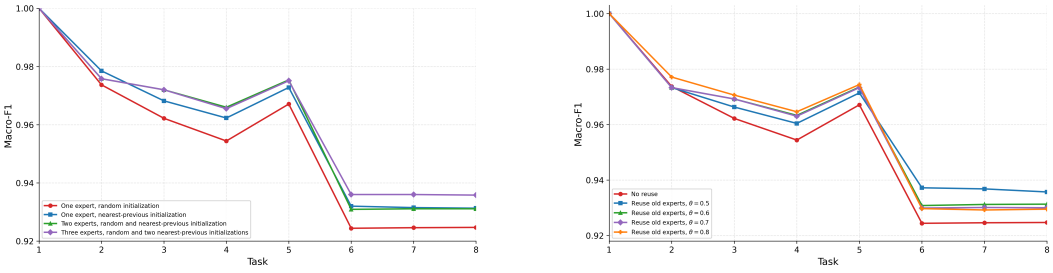
Initializing from a related previous expert improves over random initialization in Figure 7(a). With a single expert, nearest-previous initialization raises Last Macro-F1 from 0.9247 to 0.9313; using three initialization candidates, one random and two copied from related previous experts, further improves Last Macro-F1 to 0.9358.

Similarity-Guided Old-Expert Reuse. Previously learned experts can also be reused as frozen residual candidates. In this variant, the current source trains new experts, attaches selected old experts without updating them, and expands the current gate over both sets.

The old-expert set is selected before current-source expert training using source-fVAE evidence. After fitting the current source fVAE, we compare its reconstruction error e_{self} with each previous task-branch error e_k on the current images and compute

$$s_k = \exp\left(-\frac{\max(0, e_k - e_{\text{self}})}{\tau_{\text{sim}} e_{\text{self}}}\right).$$

We fix $\tau_{\text{sim}} = 20$ and reuse previous experts with $s_k \geq \theta$, sweeping $\theta \in \{0.5, 0.6, 0.7, 0.8\}$. The router is unchanged at inference; only the selected task branch may mix trained new experts with frozen reused experts. In Figure 7(b), the best setting, $\theta = 0.5$, raises Last Macro-F1 from 0.9247 to 0.9357.



(a) Expert initialization under Recon routing

(b) Similarity-guided old-expert reuse

Figure 7: Cumulative Macro-F1 after each incremental step for expert initialization and similarity-guided old-expert reuse ablations. Curves show means over three seeds.

Figure 7 shows that the best multi-candidate initialization and similarity-guided old-expert reuse reach nearly the same Last Macro-F1 (0.9358 vs. 0.9357). Since the default single-expert variant still

improves over random initialization (0.9313 vs. 0.9247), we use single nearest-expert initialization by default: it changes only the starting point, preserves the one-expert parameter budget, and avoids retaining additional frozen experts at inference.

D.3 Adapter Placement

Adapter placement controls where task-specific residual corrections enter the frozen backbone. Table 4 uses new-random expert initialization, IS routing with $N_{\mathrm{mc}} = 600$, likelihood-style scoring, and top-1 inference while varying the adapter insertion sites. For DeiT-Small, each variant inserts three residual adapter modules into transformer blocks 1–3, 4–6, 7–9, or 10–12. For ResNet-18, the shallow, middle, and deep variants insert adapters after `layer1.0`, `layer1.1`, and `layer2.0`; after `layer2.1`, `layer3.0`, and `layer3.1`; and after `layer3.1`, `layer4.0`, and `layer4.1`, respectively.

On DeiT-Small, mid1 is best; shallow, mid2, and deep are lower and close to each other. On ResNet-18, mid is also best, and shallow is clearly worse. Among these grouped placements under IS routing, middle layers work best: they provide more discriminative features than shallow layers while retaining subsequent computation after the residual update.

Table 4: Adapter position ablation on the industrial surface-defect benchmark under IS routing. Values are mean ± standard deviation over three seeds.

Backbone	Adapter position	Last mAcc	Last Macro-F1	Avg mAcc	Avg Macro-F1
DeiT-Small	shallow blocks	84.19±1.05	85.48±0.62	89.25±1.09	89.94±0.84
	mid1 blocks	**86.67±0.86**	**88.02±0.52**	**90.92±0.80**	**91.80±0.56**
	mid2 blocks	83.42±1.13	85.06±1.10	88.88±0.73	89.99±0.72
	deep blocks	83.47±0.64	85.09±0.80	88.41±0.99	89.61±0.68
ResNet-18	shallow blocks	82.43±1.59	84.14±1.69	87.42±1.66	88.73±1.81
	mid blocks	**87.13±0.25**	**88.19±0.46**	**91.13±0.39**	**91.99±0.21**
	deep blocks	85.03±2.39	86.33±2.06	89.95±1.51	90.90±1.41

D.4 Feature Reconstruction Analysis

In Table 5 the reconstructed features keep most of the frozen direction (mean cosine 0.976) and a fair amount of the variance ($R^2 = 0.785$). Classifying those reconstructions with nearest class prototypes reaches only 0.809, consistent with the fVAE being used for task selection rather than final class prediction. The low prototype accuracy for some tasks also explains why we do not use reconstructed features as classifier inputs.

Table 5: How well each task's source fVAE reconstructs frozen routing features. Prototype acc. is nearest-class accuracy on the reconstructions, not the routing accuracy.

Task	Recon MSE	Cosine similarity	R^2	Recon prototype acc.
All-task avg	0.032	0.976	0.785	0.809
Task 1	0.032	0.968	0.802	0.946
Task 2	0.050	0.953	0.722	0.870
Task 3	0.028	0.982	0.778	0.493
Task 4	0.045	0.971	0.841	0.768
Task 5	0.015	0.990	0.957	0.998
Task 6	0.036	0.969	0.768	0.565
Task 7	0.010	0.983	0.641	0.918
Task 8	0.038	0.989	0.772	0.915

Figure 8 shows fVAE-generated features for tasks 1, 3, 5, and 8, mapped to images by a separately trained feature-to-image decoder used only for visualization. The decoded images capture coarse source appearance but lack fine detail. Task routing relies on feature reconstruction errors, so the

fVAE needs to capture features that distinguish sources without the additional demands of realistic image synthesis.

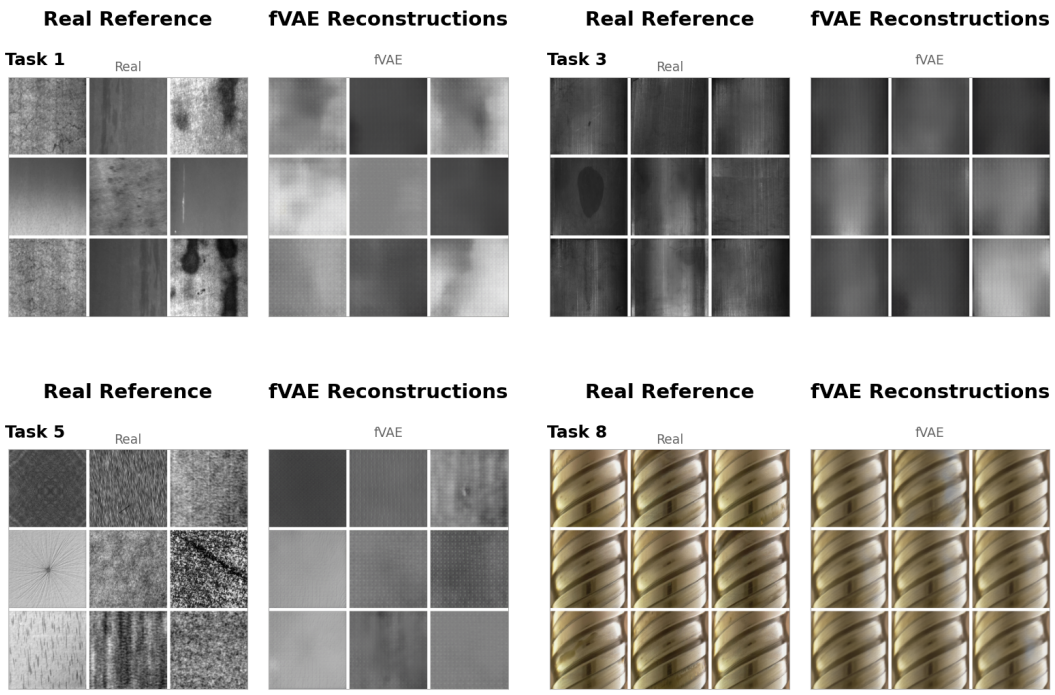


Figure 8: Real references and decoded fVAE feature samples for tasks 1, 3, 5, and 8.

D.5 Sensitivity to the Number of Reconstruction Samples

We vary the number of Monte Carlo samples used for reconstruction-based routing. Table 6 keeps the training protocol, new-random expert initialization, and task-level Recon routing fixed while changing N_{mc} at inference. This checks whether the routing decision depends on a large sampling budget.

The Last scores are stable across a wide range of N_{mc}. Even $N_{\mathrm{mc}} = 6$ obtains 92.44 ± 0.82 Last mAcc and 92.54 ± 0.66 Last Macro-F1. Increasing the budget to $N_{\mathrm{mc}} = 2000$ mainly improves the Avg metrics. These results support reconstruction-based routing with a moderate sampling budget. We therefore use $N_{\mathrm{mc}} = 600$ in the main ablations as a moderate budget that avoids tuning the method to the smallest or largest sampling setting.

Table 6: Sensitivity to the number of reconstruction samples under task-level Recon routing. Values are mean ± standard deviation over three seeds.

Reconstruction samples	Last mAcc	Last Macro-F1	Avg mAcc	Avg Macro-F1
$N_{\mathrm{mc}} = 6$	**92.44** $\pm$ 0.82	**92.54** $\pm$ 0.66	95.28 $\pm$ 0.33	95.42 $\pm$ 0.34
$N_{\mathrm{mc}} = 60$	91.61 $\pm$ 0.57	92.16 $\pm$ 0.62	94.92 $\pm$ 0.48	95.21 $\pm$ 0.41
$N_{\mathrm{mc}} = 600$	92.04 $\pm$ 0.36	92.47 $\pm$ 0.44	95.13 $\pm$ 0.24	95.39 $\pm$ 0.23
$N_{\mathrm{mc}} = 2000$	92.13 $\pm$ 0.54	92.52 $\pm$ 0.64	**95.56** $\pm$ 0.39	**95.84** $\pm$ 0.43

D.6 Top-k Inference Sensitivity

We use top-1 routing as the default in the main experiments. The optional confidence-aware top-2 rule is defined in equation 12–equation 14. For this sensitivity experiment, we report both the task-aggregated IS score $\ell_k(r) = \psi_k^{\mathrm{task,IS}}(r)$ and the default reconstruction score, varying the same margin hyperparameter α.

For Top-k inference, the trained experts are fixed, and only the test-time candidate rule changes between Top-1 and Top-2 inference. Training uses new-random expert initialization, and routing scores are estimated with $N_{\mathrm{mc}} = 600$. This tests whether uncertain routing cases benefit from evaluating both local heads instead of forcing a single early decision.

Figure 9 shows that Top-2 inference does not provide a stable or substantial gain over Top-1, even though it evaluates an additional branch on low-margin samples. The Top-2 curves remain close to Top-1 across tasks under both IS routing and the default reconstruction-based task-identity score, so we adopt the simpler Top-1 rule in the main experiments.

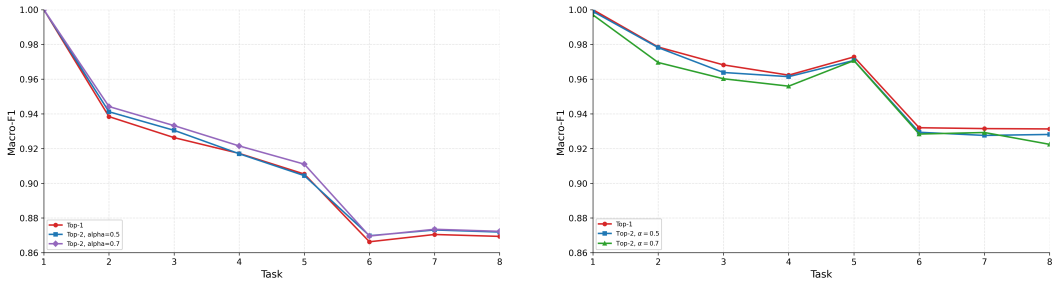


Figure 9: Cumulative Macro-F1 after each incremental step for Top-k inference under task-aggregated IS scores (left) and reconstruction scores (right). Curves show means over three seeds. Top-2 variants add the second-ranked task only when the top-1 margin is below the threshold α/K_t.

D.7 Connections Between Ablations and Theory

Head calibration. The ablation in Head Calibration without Test-Time Task Identity corresponds to the head-calibration experiment. Proposition 2 explains the information restriction of this decision rule: affine calibration selects a branch using only the branch-wise maxima $\boldsymbol{\xi}(x)$. Its routing error cannot fall below the optimal routing error based on that statistic. The observed Macro-F1 gap does not establish that this lower bound is positive, quantify it, or rule out improved calibration.

Task-identity inference. The ablation in Task-Identity Inference Strategy corresponds to the task-identity inference experiment. The prototype router indicates that frozen features contain useful source information. A mean prototype, however, provides a limited summary of a source distribution and may not capture its within-source feature variation. The fVAE variants model the same feature distributions more flexibly. Class-level fVAEs achieve higher task-identity accuracy than prototypes, although their scores still require aggregation to select a task branch.

Source-fVAE IS routing reaches 96.92 ± 0.69 Last task-ID accuracy. Reconstruction scoring reaches 99.72 ± 0.00 Last task-ID accuracy and 92.47 ± 0.44 Last Macro-F1, within one point of the oracle supplied with true task identity. At the displayed precision, Recon and Recon+KL have essentially the same task-ID accuracies and similar Last Macro-F1. Adding KL to the routing score shows no clear benefit in this comparison; the KL term remains a regularizer in fVAE training.

The Recon+KL comparison is empirical. Lemma 1 provides a sufficient condition under which expected reconstruction and marginal likelihood have the same maximizer. As detailed in Appendix B.3, the weighted KL score can retain both prior-divergence and posterior-gap terms. The analysis therefore does not imply that reconstruction-only routing is universally superior.

Source-to-task formation. The repeated-label experiment in Section 5.3.3 evaluates source-to-task assignment beyond the one-source-one-branch setting. The two main benchmarks consider the one-source-one-task case and therefore do not test whether an arriving source with a repeated label set should join an existing branch or create a new one. To examine this setting, we split the training, validation, and test partitions of `neu`, `xsdd`, `magnetic`, and `cr7` separately into two class-stratified halves. This produces eight sources with four distinct label sets, each appearing twice. Evaluation uses a unified label space: labels repeated across the two halves represent the same class, and every method is evaluated on semantic class prediction rather than source-half identification. We additionally include CN-DPM (Lee et al., 2020), a task-free expansion method that allocates mixture

experts through a Bayesian nonparametric criterion, to compare an alternative approach to capacity allocation.

The method separates source modelling from task-local classification. Methods that allocate capacity per arriving source can represent the same semantic label in several independent branches or components, retaining cross-branch score competition at inference. HiDMoA retains one source fVAE per source for routing evidence while assigning compatible repeated-label sources to a shared task branch and local cosine head. The reused branch is adapted through replay-free distillation: the update fits the new source while encouraging consistency with pre-update outputs on current-source inputs.

Corollary 2 formalizes how source-to-task aggregation removes the need to distinguish sources assigned to the same task branch. Combined with task-local adaptation, this organization supports shared classification across repeated-label sources, for which the experiment shows strong aggregate performance.

E Controlled Simulation

This appendix uses a Gaussian feature-space stream to separate two error sources: picking the wrong task, and misclassifying inside the right one. The setup is deliberately simpler than the image benchmarks, so the numbers should not be read as another benchmark result. Its role is to check the mechanism in a controlled setting: a global $\arg\max$ can still fail when local heads are reasonable, if those heads were trained on different clusters and their scores are compared after the fact.

E.1 Synthetic Stream and Models

We construct a high-dimensional Gaussian incremental stream and vary two knobs. One knob is how separable classes are inside a task. The other is how much neighboring tasks overlap. This gives cases where the local expert is the bottleneck, cases where routing is the bottleneck, and cases where both contribute. The stream has $T = 10$ tasks, each corresponding to one source with two new classes, input dimension $D_{\text{in}} = 128$, and 2000 samples per class. Samples are split into train, validation, and test sets with ratio $7 : 1.5 : 1.5$, and all results are averaged over ten independent runs.

Each class c belongs to a latent task source $\zeta(c)$. We sample task directions a_ζ and class-offset directions b_c from the unit sphere and generate

$$x \mid y = c \sim \mathcal{N}(\mu_c, \sigma^2 I_{D_{\text{in}}}), \qquad \mu_c = r_{\text{task}} a_{\zeta(c)} + r_{\text{cls}} b_c.$$

We fix $r_{\text{task}} = 4.0$ and $\sigma = 1.2$. The intra-task difficulty is set by r_{cls}: Easy uses $r_{\text{cls}} = 2.5$, Difficult uses $r_{\text{cls}} = 1.4$. Difficult mainly makes the local expert worse even when the task is known.

To control inter-task ambiguity, adjacent tasks are paired into conflict groups. Let $\bar{\mu}_t$ be the mean of task t and $\bar{\mu}_{\mathcal{G}(t)}$ the mean of its conflict group. After sampling x_i from the Gaussian stream, we apply the task-level shift

$$\tilde{x}_i = x_i + \lambda(\bar{\mu}_{\mathcal{G}(t)} - \bar{\mu}_t). \tag{15}$$

The same shift is applied to every sample in task t, so within-task class geometry is largely preserved while task centers move toward their group mean. No Conflict corresponds to $\lambda = 0$; Conflict uses $\lambda = 0.3$. This yields a 2×2 design: Easy or Difficult intra-task difficulty, crossed with No Conflict or Conflict inter-task ambiguity.

At task t, the model instantiates a task-local expert $\varphi_t : \mathbb{R}^{D_{\text{in}}} \to \mathbb{R}^{64}$, implemented as a $D_{\text{in}} \to 256 \to 64$ ReLU MLP. The expert is trained on the current class set $\mathcal{C}_t$ and frozen afterwards. Each expert has a fixed cosine prototype head. For class $c \in \mathcal{C}_t$, the prototype $m_{t,c}$ is the normalized mean of training features produced by φ_t, and the local score is

$$h_{t,c}(x) = s \cdot \frac{\varphi_t(x)^\top m_{t,c}}{\|\varphi_t(x)\|_2 \|m_{t,c}\|_2}, \qquad c \in \mathcal{C}_t,$$

with scale $s = 20$. Local experts use Adam with learning rate 10^{-3}, weight decay 10^{-4}, batch size 128, at most 30 epochs, early-stopping patience 5, and minimum validation-loss improvement 10^{-5}.

E.2 Compared Decision Rules

Oracle Identity uses the ground-truth task identity $t^\star$ and predicts within $\mathcal{C}_{t^\star}$. That is the local-head accuracy when the task is known. Global-Max drops the task tag and takes $\arg\max$ over every local head,

$$\hat{y} = \arg \max_{j \le t,\, c \in \mathcal{C}_j} h_{j,c}(x),$$

which is a single $\arg\max$ over heads that were never trained together.

Global-Gate learns a softmax gate,

$$\phi_{\text{gate}}(x) = \sum_{j \le t} \chi_j(x) \varphi_j(x), \qquad \chi(x) = \text{softmax}(W_g x),$$

and feeds the mixed feature into a global linear classifier over all seen classes. The global head is first trained on current-task labels for at most 20 epochs with patience 5, while the newly added class weights are trainable. Gate training then freezes the global head and old experts, and optimizes the gate plus the current expert using class cross-entropy as the training signal.

Global-Gate-Cal calibrates the same global linear head using balanced replay while keeping the gated MoE fixed. After gate training, calibration updates the head parameters,

$$(W_{\text{cal}}, b_{\text{cal}}) = \arg\min_{W,b} \frac{1}{|\mathcal{B}_t^{\text{cal}}|} \sum_{(x_i,y_i) \in \mathcal{B}_t^{\text{cal}}} -\log \frac{\exp((W\phi_{\text{gate}}(x_i) + b)_{y_i})}{\sum_{c \in \bigcup_{k=1}^{t} \mathcal{C}_k} \exp((W\phi_{\text{gate}}(x_i) + b)_c)}. \tag{16}$$

The calibration memory in Equation 16 is a total budget over all seen classes: $|\mathcal{B}_t^{\text{cal}}| \le 600$, sampled as uniformly as possible across $\bigcup_{k=1}^{t} \mathcal{C}_k$. We reserve 20% for calibration validation and train the head with Adam, learning rate 10^{-3}, weight decay 10^{-4}, up to 300 epochs, and patience 30.

Decoupled-Routing stores one raw-input task prototype after each task and predicts

$$\bar{x}_t = \frac{1}{|\mathcal{D}_t^{\text{train}}|} \sum_{(x_i,y_i) \in \mathcal{D}_t^{\text{train}}} x_i, \qquad \hat{t} = \arg\min_{j \le t} \|x - \bar{x}_j\|_2,$$

then classifies within $\mathcal{C}_{\hat{t}}$ using $\varphi_{\hat{t}}$ and its local cosine prototype head. Task prototypes are estimated from training data. We report class accuracy and first-stage identity accuracy for this rule.

E.3 Global Score Competition

Table 7 reports Final-step cumulative accuracy. Easy / No Conflict: Global-Gate-Cal is 0.702 ± 0.009, Decoupled-Routing-Cls is 0.892 ± 0.002, oracle is 0.926 ± 0.002. Difficult / Conflict: Global-Gate-Cal falls to 0.435 ± 0.011, Decoupled-Routing-Cls to 0.691 ± 0.006. The gap grows as more heads are added to the same $\arg\max$.

Table 7: Final-step cumulative accuracy in the controlled simulation. Values are mean±sd over ten runs.

Setting		Oracle	Global decision			Decoupled-Routing	
Intra-task difficulty	Inter-task ambiguity	Oracle Identity	Global-Max	Global-Gate	Global-Gate-Cal	Cls. Acc.	Identity Acc.
Easy	No Conflict	92.6±0.20	10.1±1.30	17.5±3.90	70.2±0.90	89.2±0.20	96.4±0.30
Difficult	No Conflict	78.6±0.60	7.5±0.90	10.2±2.40	50.7±1.00	74.7±0.40	95.1±0.30
Easy	Conflict	92.5±0.30	10.0±1.70	13.7±2.30	61.9±1.20	83.2±0.40	90.0±0.40
Difficult	Conflict	78.7±0.60	7.6±0.80	8.8±1.10	43.5±1.10	69.1±0.60	87.9±0.50

Figure 10 plots accuracy by task. Global-Max and the uncalibrated Global-Gate fall as heads accumulate, showing that the failure is progressive rather than confined to the last task. Global-Gate-Cal is better calibrated, but it still stays below Decoupled-Routing-Cls. This indicates that replay-based calibration improves the global-gate classifier, while separate task routing yields higher classification accuracy in this comparison. Conflict widens the gap to Oracle Identity because neighboring tasks are harder to tell apart.

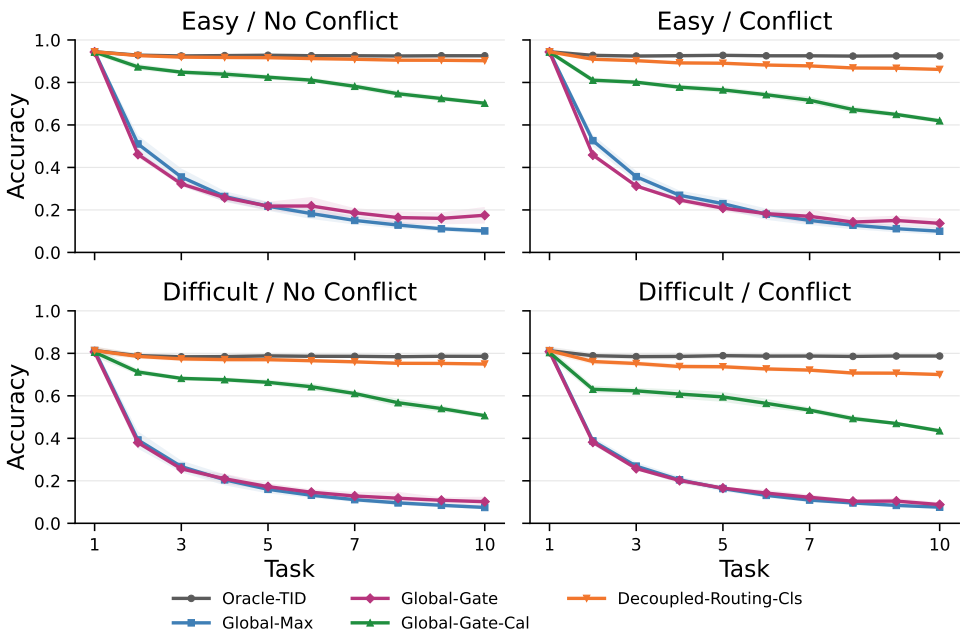


Figure 10: Accuracy over incremental tasks. Shaded regions indicate ± 1 standard deviation across ten runs.

E.4 Error Decomposition

We decompose final-step cumulative errors in Figure 11. This is mainly a diagnostic check for the simulation: the routing method should lose points from identity mistakes and local mistakes, not from an unrelated artifact of the evaluation. Let $\hat{t}_i$ and $\hat{y}_i^R$ be the task and class predictions of Decoupled-Routing, and let $\hat{y}_i^{\text{base}}$ be the prediction of a global baseline, either Global-Max or Global-Gate-Cal. t_i denotes the true task identity. With empirical probability $\widehat{\Pr}_{\mathcal{S}}$ on the test set, define

$$\begin{aligned} \text{E1} &= \widehat{\Pr}_{\mathcal{S}}[\hat{t}_i \neq t_i], \\ \text{E2} &= \widehat{\Pr}_{\mathcal{S}}[\hat{t}_i = t_i,\ \hat{y}_i^R \neq y_i], \\ \text{E3}-\text{base} &= \widehat{\Pr}_{\mathcal{S}}[\hat{y}_i^{\text{base}} = y_i,\ \hat{y}_i^R \neq y_i], \\ \text{E4}-\text{base} &= \widehat{\Pr}_{\mathcal{S}}[\hat{y}_i^{\text{base}} \neq y_i,\ \hat{y}_i^R = y_i]. \end{aligned} \tag{17}$$

In Equation 17, E1 measures identity errors, E2 measures local classification errors after correct routing, and E4 larger than E3 means that Decoupled-Routing corrects more baseline errors than it introduces.

E1 and E2 are the empirical counterparts of e_{route} and $\varepsilon_{\text{local}}$ in equation 9 for this disjoint-label simulation. Their sum is the error rate of the routed predictor. E3 and E4 compare the routed predictor with a global baseline and are not terms in that decomposition.

Conflict raises E1, which is what moving the task centers was meant to do. Difficult raises E2: classes inside a task overlap more, so the local expert still fails after a correct route. The E3/E4 comparison shows whether decoupled routing trades one error type for another. E4-GateCal is much larger than E3-GateCal in every cell, so Decoupled-Routing fixes a large share of the calibrated gate's remaining mistakes while introducing relatively few new errors.

E.5 Sensitivity to Inter-Task Conflict

We fix intra-task difficulty to Difficult and vary the conflict strength λ in Equation 15 over $0, 0.1, 0.2, 0.3, 0.4$. Figure 12 reports the final-step cumulative trend.

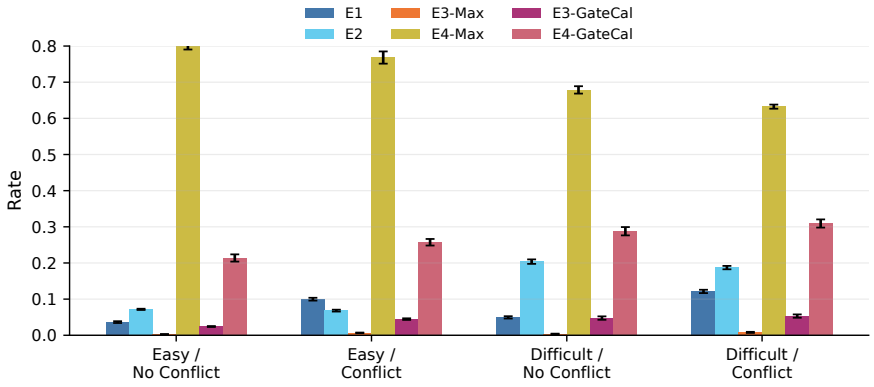


Figure 11: Final-step cumulative error split. Conflict raises identity error (E1). Difficult raises local error after a correct route (E2). E4 stays much larger than E3 against both global baselines.

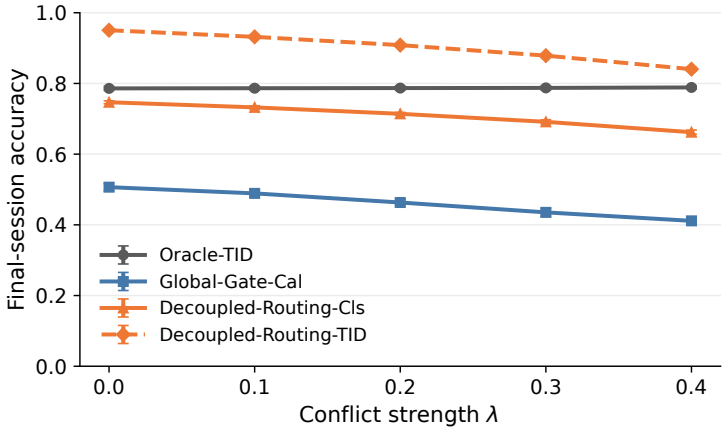


Figure 12: Sensitivity to inter-task conflict strength. As λ increases, Decoupled-Routing-Identity and Decoupled-Routing-Cls both decline. Decoupled-Routing-Cls remains above Global-Gate-Cal for all tested values.

As λ increases from 0 to 0.4, Decoupled-Routing-Identity drops from 0.951 ± 0.003 to 0.840 ± 0.005, and Decoupled-Routing-Cls drops from 0.747 ± 0.004 to 0.662 ± 0.006. Global-Gate-Cal also degrades, from 0.507 ± 0.010 to 0.411 ± 0.009. The class-accuracy gap of Decoupled-Routing-Cls over Global-Gate-Cal remains around 0.24–0.26 across the sweep.

F Dataset Statistics and Additional Metrics

F.1 Dataset Statistics

This appendix reports additional class-balanced mAcc results and the dataset sizes used to construct the incremental streams in section 5. For the two main benchmarks, each source is one data source and class labels are disjoint across sources. The repeated-label Task Formation study instead uses eight sources generated from four datasets, with two sources sharing the same label set for each underlying task. The train split is used for incremental optimization, the validation split is used for early stopping and model selection within each source, and the test split is held out for reporting class-balanced metrics after each incremental step.

Table 8 is the order-1 industrial stream used in the main tables. The other two industrial source orders keep the same eight sources and the same splits, and only permute the arrival order. Table 9

is the general stream from Table 10. It is larger, and class counts and sample counts vary more. Source sizes are unbalanced, as they are in the original datasets, so we report class-balanced mAcc and Macro-F1.

All images are resized to 224×224. Most sources use an approximate 70/15/15 split, except that CIFAR10 and Food-11 follow predefined partitions; for CIFAR10, the 10k test batch is evenly divided into validation and test subsets. Image-benchmark results use the same three seeds as section 5.

For the repeated-label stream in subsubsection 5.3.3, the train, validation, and test partitions of `neu`, `xsdd`, `magnetic`, and `cr7` are each split class-stratified into two complementary 50/50 halves. This yields eight sources and four repeated label sets. The split itself uses fixed seed 42 independently of the training seed, and test evaluation accumulates all seen source halves.

mAcc is the unweighted average class recall, $\mathrm{mAcc} = |\mathcal{C}_{1:t}|^{-1} \sum_{c \in \mathcal{C}_{1:t}} TP_c/N_c$, equivalent to balanced accuracy in the multi-class case. Macro-F1 averages per-class F1, $\text{Macro-F1} = |\mathcal{C}_{1:t}|^{-1} \sum_{c \in \mathcal{C}_{1:t}} 2P_cR_c/(P_c + R_c)$, where $P_c = TP_c/(TP_c + FP_c)$ and $R_c = TP_c/N_c$. Here N_c is the number of evaluation samples with true class c.

Table 8: Multi-source industrial surface-defect benchmark statistics for order 1.

Source	Dataset	Classes	Train	Val	Test	Total
1	neu	6	1260	270	270	1800
2	xsdd	7	954	203	203	1360
3	magnetic	6	940	202	202	1344
4	cr7	7	2898	621	621	4140
5	dagm	10	11276	2412	2412	16100
6	gc10	10	1618	347	347	2312
7	kolek	2	2334	499	503	3336
8	bsd	2	772	165	168	1105
Total	–	50	22052	4719	4726	31497

Table 9: General multi-source benchmark statistics.

Source	Dataset	Classes	Train	Val	Test	Total
1	CIFAR10	10	50000	5000	5000	60000
2	DeepWeeds	9	12256	2626	2627	17509
3	EuroSAT	10	18900	4050	4050	27000
4	Fashion-MNIST	10	49000	10500	10500	70000
5	MNIST	10	49001	10500	10499	70000
6	Oxford-IIIT Pet	37	5173	1109	1108	7390
7	PatternNet	38	21280	4560	4560	30400
8	Food-11	11	9866	3430	3347	16643
Total	–	135	215476	41775	41691	298942

F.2 General Multi-Source Results and Industrial Surface-Defect Stream Results

The general stream combines eight public datasets with differing class counts and class imbalance. Accordingly, Table 10 reports cumulative Macro-F1 after each arriving source and its average across the eight incremental steps. The DeiT-Small variant of HiDMoA achieves 93.3% Avg Macro-F1, close to TPL's 93.6%. The ResNet-50 variant raises Avg Macro-F1 to 95.4% and achieves the highest score at seven of the eight evaluation steps.

These datasets span natural images, remote sensing, handwritten digits, clothing, pets, and food. HiDMoA with DeiT-Small remains competitive with TPL, while its ResNet-50 variant improves over the DeiT-Small variant at every reported evaluation step. The results support the applicability of source-fVAE routing and task-local classification beyond industrial surface-defect recognition and show higher performance with the ResNet-50 configuration on this benchmark.

The corresponding industrial surface-defect mAcc results are reported in Table 11. HiDMoA achieves the best Last and Avg mAcc for all three source orders, with Last scores of 93.0 ± 0.4,

Table 10: General multi-source stream. Each dataset column reports cumulative Macro-F1 after learning that source; Avg is the mean over the eight incremental steps. Values are mean ± standard deviation over three seeds; the best result in each column is bold.

Method	CIFAR10	DeepWeeds	EuroSAT	Fashion MNIST	MNIST	Oxford Pet	PatternNet	Food11	Avg Macro-F1
MORE	93.8±0.10	90.3±0.30	91.5±0.40	90.8±0.10	91.4±0.10	90.1±0.20	92.2±0.50	91.8±0.40	91.5±0.10
TPL	**95.7±0.10**	91.6±0.30	93.6±0.30	93.2±0.20	94.4±0.20	92.1±0.20	94.3±0.10	93.8±0.10	93.6±0.10
Generative Classifier	85.5±0.40	77.6±0.70	82.9±0.50	84.2±0.30	86.6±0.30	86.8±0.30	89.9±0.20	89.2±0.10	85.3±0.30
DER	78.0±13.00	78.4±5.90	79.2±3.40	79.6±1.80	82.1±4.00	53.5±4.60	64.9±2.20	52.9±4.90	71.0±4.80
PEC	38.0±1.60	23.7±1.00	38.6±1.00	45.8±1.30	50.7±1.70	29.8±0.90	40.3±3.90	38.2±3.90	38.1±0.50
SEED	94.5±0.10	90.2±1.00	81.3±7.50	64.2±5.50	55.1±6.30	55.0±7.80	47.0±9.70	42.3±9.30	66.2±5.50
SEMA	95.3±0.20	50.5±7.60	45.6±6.30	44.0±4.40	39.5±7.30	51.7±1.80	61.3±5.00	61.5±2.00	56.2±2.60
HiDMoA-DeiT	92.6±0.23	92.0±0.13	93.7±0.08	93.3±0.21	94.1±0.16	92.1±0.38	94.2±0.31	94.1±0.29	93.3±0.20
HiDMoA-ResNet-50	95.1±0.20	**94.7±0.30**	**95.7±0.10**	**95.1±0.10**	**95.7±0.10**	**94.6±0.10**	**96.1±0.00**	**95.9±0.00**	**95.4±0.10**

92.8 ± 0.3, and 92.5 ± 0.6. This consistent performance across source orders supports the robustness of the method on the industrial stream.

Table 11: Industrial surface-defect stream mAcc results corresponding to Table 1. Last is after all sources; Avg is the mean over sources. Values are mean ± standard deviation over three seeds; best result per column is bolded.

Method	Order 1		Order 2		Order 3	
	Last mAcc	Avg mAcc	Last mAcc	Avg mAcc	Last mAcc	Avg mAcc
BUILD	24.6±4.10	61.8±4.80	31.4±2.60	60.3±8.40	43.5±5.10	67.6±0.50
MORE	84.2±1.20	87.3±1.10	83.8±0.10	88.9±0.50	84.4±0.50	87.4±0.20
TPL	85.1±0.40	90.4±0.40	84.2±0.60	88.8±0.30	83.8±0.60	90.7±0.50
Generative Classifier	83.6±0.30	88.0±0.40	83.5±0.40	88.9±0.40	83.9±0.50	88.4±0.50
DER	41.5±10.20	75.8±1.20	52.2±7.60	75.2±2.40	56.1±6.50	77.0±0.90
PEC	65.0±1.10	68.5±0.90	66.8±1.10	76.3±0.90	65.4±1.40	64.2±1.80
SEED	72.3±0.60	79.9±0.10	72.1±0.50	82.3±0.40	71.3±0.50	77.3±1.20
SEMA	75.9±2.50	81.7±2.20	71.8±3.40	82.8±6.00	75.5±1.80	85.2±0.90
TagFex	76.3±0.20	71.6±0.60	77.3±1.40	84.5±0.80	53.7±14.30	71.8±7.20
HiDMoA	**93.0±0.40**	**96.0±0.40**	**92.8±0.30**	**94.6±0.40**	**92.5±0.60**	**95.9±0.40**

G Implementation Protocol

Implementation. The main-text experiments use a frozen DeiT-Small classification backbone (Touvron et al., 2021), with residual adapters after blocks 8, 10, and 12 and local cosine heads initialized by class-mean imprinting. Each source fVAE is fitted to its source's features from a separate frozen ResNet-18 routing encoder (He et al., 2016), and source-fVAE scores are grouped by task branch at inference. Main results use reconstruction-based top-1 routing; IS and Recon+KL serve as routing ablation variants. We additionally evaluate ResNet-18 for adapter placement and ResNet-50 as the stronger-backbone variant on the general benchmark. Task branches are trained with Adam using learning rate 10^{-3}, weight decay 10^{-4}, batch size 32, and early stopping. Results are reported as mean ± standard deviation over seeds $\{42, 43, 44\}$. Last denotes the result after the final source, and Avg denotes the mean of these cumulative results across incremental steps.

HiDMoA configuration. The classification backbone F is frozen throughout. The default DeiT-Small variant inserts one adapter per adapted stage, $M_\ell = 1$, after blocks 8, 10, and 12, using linear down and up projections with LayerNorm, and its readout U is the final normalization and pre-logits with class-token readout. The ResNet variants instead use gated mixtures with $M_\ell > 1$ and 1×1 convolutional adapters with BatchNorm, and take U to be global average pooling; adapter placement is ablated in Appendix D. Local heads are cosine heads with class-mean-imprinted prototypes. The routing encoder G is a separate frozen ResNet-18, so the routing space is fixed before the first source. Each source fVAE is trained with $\lambda_{\text{rec}} = 1.0$ and $\beta = 0.01$ in equation 1 without KL warmup, the

Task Formation ratio uses $\epsilon = 10^{-12}$, and routing scores use $N_{\mathrm{mc}} = 600$ posterior samples in equation 2; Appendix D reports similar Last metrics at smaller sampling budgets.

Task formation protocol. The source-to-task-branch compatibility threshold is fixed globally at $\gamma_{\mathrm{TF}} = 5$ and is not tuned per task or per source. Merge updates use $T_{\mathrm{kd}} = 1$ and search $\lambda_{\mathrm{kd}} \in \{0, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10\}$. For each candidate λ_{kd}, the matched task branch is restored to the same pre-merge snapshot, probed for five epochs on the current source training set without early stopping, and evaluated only on the current source validation set. We select the validation-accuracy maximizer, restore the pre-merge snapshot again, and perform the full update with the selected coefficient using the same learning rate and early-stopping rule as ordinary task-branch training. Distillation is applied only on current-source training inputs; previous-source train and validation samples are neither replayed nor revisited. Every source retains its source fVAE V_t, indexed by the source identifier and registered under the assigned task through $\mathcal{S}_k$.

For reproducibility, Table 12 reports the implementation protocol used for external baselines in our benchmark. All methods are evaluated on the same source orders, train/validation/test splits, image resolution, and seed set. Replay memory, OOD selection, expert routing, and expansion modules follow the original methods.

Several baselines have multi-stage procedures, so we report the training stage lengths and early-stopping patience used in our runs. A memory budget of 600 denotes the maximum total number of retained exemplars or replay samples. The memory is class-balanced, with the available budget distributed approximately uniformly across previously seen classes. Replay-free methods are marked as "none" in the memory column.

Table 12: External baseline implementation protocol. "ES" denotes early-stopping patience.

Method	Backbone/ model	Memory/ replay	Batch	Main epochs	Learning rate	ES / notes
BUILD	DeiT-Small	none	32	40	5×10^{-3}	ES 15; post-hoc OOD task selection
MORE	DeiT-Small	memory budget 600	32	40	10^{-3}	ES 15; OOD replay/back-update enabled
TPL	DeiT-Small	memory budget 600	32	40	10^{-3}	ES 15; SGD-HAT, cosine scheduler
Generative Classifier	ResNet-18	none	64	100	10^{-3}	ES 10; 600 IS samples
DER	ResNet-18	memory budget 600	32	80	5×10^{-2}	ES 15; SGD-style PyCIL recipe
PEC	PEC CNN	none	8	40	10^{-3}	Adam; linear scheduler
SEED	ResNet-18	memory budget 600	32	100	10^{-2}	ES 20; momentum 0.9
SEMA	ViT-B/16	none	32	100	5×10^{-3}	ES 30; cosine scheduler
TagFex	ResNet-18	memory budget 600	32	80	10^{-2}	SGD; official two-stage optimizer settings

H Efficiency and Resource Analysis

We report resource usage on the industrial surface-defect benchmark under source order 1 in Table 13. Forward FLOPs are the cost of one input. Pipeline FLOPs also count routing or generation stages recorded by each method's logger.

For HiDMoA, the reported cost corresponds to the one-expert initialization setting. On the disjoint-label industrial stream, every source starts a new task branch, so these numbers coincide with the original one-branch-per-source implementation. In the general Task Formation formulation, task-branch storage grows with the number of task branches K_t, whereas one source fVAE is retained per source. A merged source therefore avoids another task branch but still contributes one source fVAE. The full model has 45.84M parameters, trains 11.80M of them, and uses 0.91GB peak memory. Pipeline FLOPs are 9766T, lower than most baselines.

Table 13: Resource usage on the eight-source industrial surface-defect stream under source order 1, averaged over three seeds.

Method	P_{total} (M)	P_{train} (M)	GPU (GB)	FLOPs/ forward (G)	FLOPs/ pipeline (T)
BUILD	32.87	9.65	2.18	71.58	17538.73
MORE	23.13	1.46	4.06	71.71	50164.50
TPL	23.25	1.58	6.71	71.70	51476.16
Gen. Classifier	73.58	62.40	0.99	39.10	720.80
DER	89.63	89.62	1.82	29.18	88715.95
PEC	6.02	5.57	2.07	6.95	1042.00
SEED	90.59	89.55	1.66	29.34	15705.82
SEMA	91.97	5.41	1.85	22.80	171607.09
TagFex	105.46	105.46	7.98	110.54	355736.39
HiDMoA	45.84	11.80	0.91	18.37	9765.91

I LIMITATIONS

Limitations. The current method has several limitations. First, routing quality depends on whether different sources remain sufficiently separable in the frozen routing feature space. When sources are highly similar or their distributions overlap strongly, reconstruction scores can become ambiguous, which may require stronger routing features, calibrated multi-branch inference, or uncertainty-aware decisions.

As analyzed in Appendix B.3, a small positive reconstruction margin may require a larger sampling budget under Corollary 4. More samples reduce Monte Carlo error, but cannot correct an expected-score ranking that favors an incorrect source or an inadmissible branch. The shared frozen metric alone guarantees neither a positive margin nor correct routing. Such failures contribute to the routing term of equation 9. Second, Task Formation reduces task-branch growth from the number of sources to the number of task branches, but the current implementation still retains one source fVAE for every observed source. Consequently, routing storage and test-time scoring cost continue to increase with the number of sources, even when several sources share one task branch. Third, the current assignment rule is deliberately conservative: it allows a source to merge only when its label set exactly matches the label set of an existing task branch, $\mathcal{C}_t = \mathcal{Y}_k$. A source with only partially overlapping classes is therefore not handled by the present merge rule. Finally, the method assumes known source boundaries during training and does not yet address fully boundary-free online streams.