CONDITIONAL COMPLETION RANKING FOR TABULAR SELF-SUPERVISED LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Self-supervised learning (SSL) for tabular data aims to capture dependencies among heterogeneous features. However, existing reconstruction and view-alignment objectives often emphasize target prediction or instance-level invariance, which may not fully capture the complex, multi-modal relationships among tabular features. We instead formulate tabular SSL as learning the *conditional-to-marginal shift*, which measures how the probability of a multi-column candidate completion changes after conditioning on the observed features relative to its marginal probability. Based on this formulation, we propose Conditional Completion Ranking (**CCR**), which distinguishes samples from the joint distribution from samples drawn from the product of marginals. We show that the optimal CCR score recovers the log conditional-to-marginal density ratio, without explicitly modeling the full conditional distribution. Experiments on 50 datasets from two benchmarks show that CCR achieves the best average rank in the few-shot settings under both frozen and fine-tuned evaluation. The code is available in the supplementary materials.

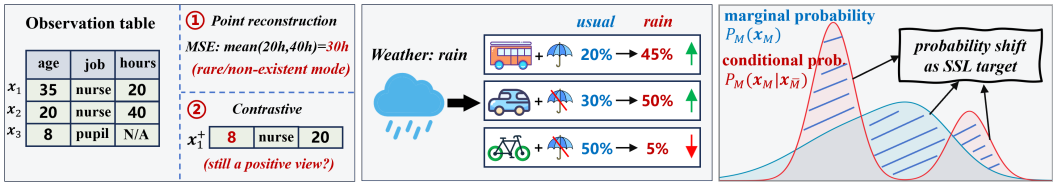
1 INTRODUCTION

Self-supervised learning (SSL) has greatly advanced representation learning by exploiting large-scale unlabeled data, achieving remarkable progress in computer vision, natural language processing, and audio analysis (Zhang et al., 2025). The success of SSL in these domains benefits from their intrinsic topological priors and local smoothness (Wittscher, 2026). Visual patches and audio frames possess inherent spatial or temporal contiguity, allowing reconstruction-based paradigms (e.g., masked autoencoders (He et al., 2022)) to smoothly interpolate missing signals based on their surroundings. Similarly, continuous perceptual spaces exhibit strong semantic invariance to augmentations—a cropped or color-jittered image of a dog remains a dog—enabling contrastive paradigms to reliably align positive views and extract robust representations without destroying the underlying semantics (Chen et al., 2020).

However, directly transferring these paradigms to tabular data introduces an inductive mismatch. Unlike perceptual signals, tabular features are often heterogeneous, unordered, and non-continuous, with dependencies that may follow discrete or multi-valued patterns rather than smooth transitions. Under such structures, reconstruction-based methods (e.g., VIME (Yoon et al., 2020)) learn by recovering observed values. For instance, if employees in a given role typically work either 20 or 40 hours per week with similar probabilities, the expected MSE is minimized by predicting 30 hours—a compromise that represents neither mode. Conversely, contrastive methods such as SCARF (Bahri et al., 2022) enforce invariance by aligning an augmented view with the original row, yet tabular semantics can be highly sensitive to feature-level perturbations: altering a single categorical value (e.g., medical diagnosis) can fundamentally disrupt cross-column relationships and render the prescribed positive pair semantically inconsistent (Lee et al., 2025). Although these failures manifest differently, they point to a common problem: point reconstruction and prescribed view invariance may not adequately capture the complex, multi-modal relationships among tabular features.

Characterizing such multi-modal relationships involves understanding the distribution of one feature or feature subset given the others, that is, its conditional distribution. Unfortunately, directly modeling the full conditional distribution is difficult, particularly for heterogeneous data with multivariate or multimodal targets (Sugiyama et al., 2010). From an information-theoretic perspective, the in-

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



(a) Objective mismatch (b) Rainy-day example (c) Conditional-to-marginal shift

Figure 1: Motivation of this work. (a) Point reconstruction may average plausible modes, while contrastive corruption may create an inconsistent positive view. (b) A rainy-day example: how context (rain) changes the plausibility of different outcomes. (c) This change is formalized as the shift from marginal to conditional probability.

formation provided by an observation is reflected in how it changes the uncertainty about another variable (Shannon, 1948; Cover and Thomas, 2006). Mutual information expresses this uncertainty reduction as the expected log-ratio of conditional to marginal probabilities, linking it to changes in each candidate's plausibility. This motivates us to model the *change* itself: how the observed context increases or decreases each candidate's plausibility relative to its marginal frequency.

Fig.1 traces our motivation. Fig.1(a) shows why deterministic prediction and prescribed view invariance can be ill-suited to discrete or multimodal tabular relationships: point reconstruction averages plausible 20-hour and 40-hour work patterns into a 30-hour compromise that represents neither mode, while marginal feature swapping can produce semantically implausible positive pairs, such as an 8-year-old nurse. These failures motivate an objective that, rather than committing to a single prediction or prescribed match, captures how context redistributes plausibility across alternative combinations of the remaining features. In the rainy-day example in Fig.1(b), observing rain increases the probabilities of taking a bus or driving and decreases that of cycling relative to their marginal probabilities, while leaving multiple outcomes plausible. Fig.1(c) formalizes this comparison through the marginal distribution $p_M(x_M)$ and the conditional distribution $p_M(x_M \mid x_{\bar{M}})$, allowing the same context to increase the probabilities of multiple completions. We refer to this change in a completion's probability relative to its marginal probability as the *conditional-to-marginal shift*.

A straightforward way to learn this shift is to estimate the normalized conditional and marginal distributions separately and then take their ratio. This approach is particularly challenging in heterogeneous data spaces, and errors in the two density estimates may compound in their ratio (Sugiyama et al., 2007; Suzuki et al., 2008). Yet the shift is itself a relative quantity, allowing density-ratio estimation to target it directly without recovering either distribution separately (Sugiyama et al., 2012). We implement this idea through binary source discrimination: observed contexts and completions are sampled jointly, whereas independently recombined alternatives follow the product of their marginals (Gutmann and Hyvärinen, 2010). We show that the Bayes-optimal logit recovers the conditional-to-marginal log-density ratio, ranking candidate completions by how strongly the context supports them relative to their marginal probabilities. To this end, we propose Conditional Completion Ranking (**CCR**), a self-supervised framework for learning context-dependent compatibility among tabular feature subsets.

The primary contributions of this work are summarized as follows:

- **Conditional-to-Marginal Formulation:** We formulate masked tabular SSL as learning how the observed columns shift the plausibility of a multi-column completion relative to its marginal frequency. We show that the optimal CCR score recovers this conditional-to-marginal log-density ratio without explicitly estimating the conditional distribution.
- **Conditional Completion Ranking:** We propose Conditional Completion Ranking (CCR), a self-supervised framework that learns context-dependent compatibility by comparing observed completions with alternatives sampled jointly from the marginal completion distribution under the same context. By varying feature partitions during pretraining, CCR trains a shared encoder to capture dependencies among different feature subsets.
- **Extensive Empirical Evaluation:** We evaluate CCR on 50 tabular classification datasets from OpenML-CC18 and TabArena. CCR achieves the best overall average rank under both frozen and fine-tuned few-shot protocols, while further analysis experiments validate its mechanism.

2 Related Work

2.1 Self-supervised Learning for Tabular Data

Self-supervised learning (SSL) constructs training signals from unlabeled data and has become a widely used approach to representation learning (Gui et al., 2024). Focusing on single-table SSL, modeling the full conditional distribution is challenging because columns are heterogeneous and their dependencies vary across datasets (Wang et al., 2025b). Existing methods can be broadly divided into predictive and contrastive approaches.

Predictive methods mainly construct supervision through reconstruction tasks. VIME (Yoon et al., 2020) and TabNet (Arik and Pfister, 2021) recover corrupted feature values, with VIME also identifying which features were corrupted; SubTab (Ucar et al., 2021) reconstructs the full input from different feature subsets and further encourages consistency across the resulting representations. SwitchTab (Wu et al., 2024) disentangles and exchanges different components of the latent representation before reconstructing the input. Despite their different designs, these methods mainly supervise the encoder through reconstruction errors in the original feature space. When the conditional distribution of a numerical target is multimodal, squared-error reconstruction targets the conditional mean, which may lie between plausible modes. T-JEPA (Thimonier et al., 2025) moves masked prediction to the latent space, but its ℓ_2 point-prediction loss may still average plausible latent targets under multimodality.

Contrastive methods learn tabular representations by aligning different views constructed from the same sample. For example, SAINT (Somepalli et al., 2021) uses CutMix and Mixup to generate positive views and contrasts them against other samples in the batch. SCARF (Bahri et al., 2022) replaces a subset of features with values sampled from their empirical marginal distributions and then aligns the corrupted view with the original one. Despite differences in architecture and corruption strategy, both methods perform contrastive learning at the sample level without explicitly characterizing distributional relationships among columns. More recent methods, such as SleepFM (Thapa et al., 2024) and ICE-T (Tokar and Sanner, 2025), contrast a modality or a single column with the average representation of the remaining parts to learn cross-column distributional relationships. However, this single-feature vs. rest formulation does not directly train relationships between a context comprising multiple observed parts and a completion comprising multiple held-out parts.

More generally, predictive and view-alignment methods construct "labels" from observed targets or prescribed view correspondences. Although such labels provide useful supervision, these objectives do not directly evaluate how context changes the relative support for joint completions. This distinction becomes important when tabular dependencies are discrete and non-smooth: small context changes can abruptly alter compatibility while multiple completions remain plausible.

2.2 Density-Ratio Estimation

Density-ratio estimation compares the relative likelihood of samples under two distributions without separately estimating their densities (Sugiyama et al., 2012). Noise-contrastive estimation (NCE) introduced a classification-based approach to this problem by distinguishing observed data from samples drawn from a reference noise distribution (Gutmann and Hyvärinen, 2010).

Density-ratio estimation has been applied to a broad range of problems, including reinforcement learning (Chen and Jiang, 2022), simulation-based inference (Delaunoy et al., 2022), covariate-shift adaptation (Xu et al., 2025), and high-dimensional distribution comparison (Wang et al., 2025a). In SSL, Oord et al. (2018) use InfoNCE to identify matched future observations among independently sampled candidates, while Deep InfoMax (Hjelm et al., 2018) distinguishes samples from a joint distribution and the product of its marginals to capture statistical dependence. Tosh et al. (2021) show that distinguishing observed from independently recombined pairs recovers the joint-to-product-of-marginals log-density ratio, and relate the representations to downstream prediction.

3 Problem Statement

Setup and Notation. Consider an unlabeled tabular dataset $\mathcal{D} = \{\boldsymbol{x}_i\}_{i=1}^N$ drawn i.i.d. from a data distribution $p(\boldsymbol{x})$ over a d-column heterogeneous feature space $\mathcal{X} = \prod_{j=1}^d \mathcal{X}_j$. Each column may

be numerical or categorical. We sample a non-empty subset of target columns $M \subsetneq \{1, \ldots, d\}$ according to a masking scheme q, and denote its complement by $\bar{M}$. For a fixed mask M, a row $\boldsymbol{x}$ is decomposed into an observed context $\boldsymbol{u} = \boldsymbol{x}_{\bar{M}} \in \mathcal{X}_{\bar{M}}$ and a target completion $\boldsymbol{v} = \boldsymbol{x}_M \in \mathcal{X}_M$. Let $p_M(\boldsymbol{u}, \boldsymbol{v})$ denote their joint distribution, with $p_M(\boldsymbol{u})$ and $p_M(\boldsymbol{v})$ denoting the corresponding marginals. Throughout, p_M denotes a density or probability mass function as appropriate for the underlying feature types.

Let $E_\theta : \mathcal{X} \to \mathcal{H}$ denote the encoder retained after self-supervised pretraining. The goal of tabular SSL is to learn from unlabeled data a representation $\boldsymbol{h} = E_\theta(\boldsymbol{x})$ that captures regularities useful across downstream tasks while reducing sensitivity to incidental variations. Since downstream supervision is unavailable during pretraining, this goal is pursued through a pretext objective constructed from the input table itself.

Reframing SSL as a Conditional-to-Marginal Shift Problem. To construct the SSL objective, we focus on the statistical relationship between the observed and target feature subsets. For each mask M, this relationship is described by the conditional distribution $p_M(\boldsymbol{v} \mid \boldsymbol{u})$. However, the conditional probability alone does not isolate the contribution of the current context. A candidate may have high conditional probability because it is likely under the overall data distribution, because it is strongly supported by the current context, or both. The conditional log-probability is given by:

$$\log p_M(\boldsymbol{v} \mid \boldsymbol{u}) = \log p_M(\boldsymbol{v}) + \log \frac{p_M(\boldsymbol{v} \mid \boldsymbol{u})}{p_M(\boldsymbol{v})} \tag{1}$$

Here, $\log p_M(\boldsymbol{v})$ describes the candidate under the marginal completion distribution and is independent of the current context. The second term captures the change in its probability after conditioning on $\boldsymbol{u}$. Motivated by this decomposition, we introduce $r_M(\boldsymbol{u}, \boldsymbol{v})$ to denote the conditional-to-marginal probability shift in Eq. (1):

$$r_M(\boldsymbol{u}, \boldsymbol{v}) = \log \frac{p_M(\boldsymbol{v} \mid \boldsymbol{u})}{p_M(\boldsymbol{v})} = \log \frac{p_M(\boldsymbol{u}, \boldsymbol{v})}{p_M(\boldsymbol{u}) p_M(\boldsymbol{v})} \tag{2}$$

For a fixed mask M, $r_M(u, v)$ is the pointwise mutual information between $\boldsymbol{u}$ and $\boldsymbol{v}$. This quantity measures how the probability of $\boldsymbol{v}$ changes after observing $\boldsymbol{u}$, relative to its global marginal probability. It is not a one-sided importance score: both large positive and large negative values can carry useful structural information. The value of $r_M(\boldsymbol{u}, \boldsymbol{v})$ has three interpretations:

- $r_M(\boldsymbol{u}, \boldsymbol{v}) > 0$: The context increases the probability of the completion, i.e., $p_M(\boldsymbol{v} \mid \boldsymbol{u}) > p_M(\boldsymbol{v})$. This indicates that $\boldsymbol{v}$ is more compatible with the context than its marginal frequency suggests.
- $r_M(\boldsymbol{u}, \boldsymbol{v}) \approx 0$: The context does not substantially change the probability of the completion, i.e., $p_M(\boldsymbol{v} \mid \boldsymbol{u}) \approx p_M(\boldsymbol{v})$. In this case, the observed context provides little additional information about this candidate beyond its global frequency.
- $r_M(\boldsymbol{u}, \boldsymbol{v}) < 0$: The context decreases the probability of the completion, i.e., $p_M(\boldsymbol{v} \mid \boldsymbol{u}) < p_M(\boldsymbol{v})$. This negative shift is also informative, because it reveals feature configurations that are less likely under the current context than under the marginal distribution.

Learning Objective. The conditional-to-marginal probability shift $r_M(\boldsymbol{u}, \boldsymbol{v})$ specifies the statistical signal that we aim to capture during self-supervised pretraining. Our goal is to learn a context encoder E_θ whose representation $\boldsymbol{h}_\theta(\boldsymbol{u}; M)$ retains the information needed to evaluate this shift for different candidate completions. Given the context representation and a candidate $\boldsymbol{v}$, the model should be able to estimate how the probability of $\boldsymbol{v}$ changes relative to its marginal probability. However, $r_M(\boldsymbol{u}, \boldsymbol{v})$ depends on unknown population distributions and cannot be used directly as a supervision target. We will introduce a binary discrimination objective that learns this conditional-to-marginal shift directly from observed data in the next section.

4 Proposed Method

To learn the conditional-to-marginal shift from observed data, we propose *Conditional Completion Ranking* (CCR). As illustrated in Fig. 2, the key idea is to transform conditional completion modeling into a binary discrimination problem. Given an observed context, CCR compares its observed completion with completions sampled from the global marginal distribution. This turns masked completion modeling into a discriminative scoring problem: the model is trained to identify whether

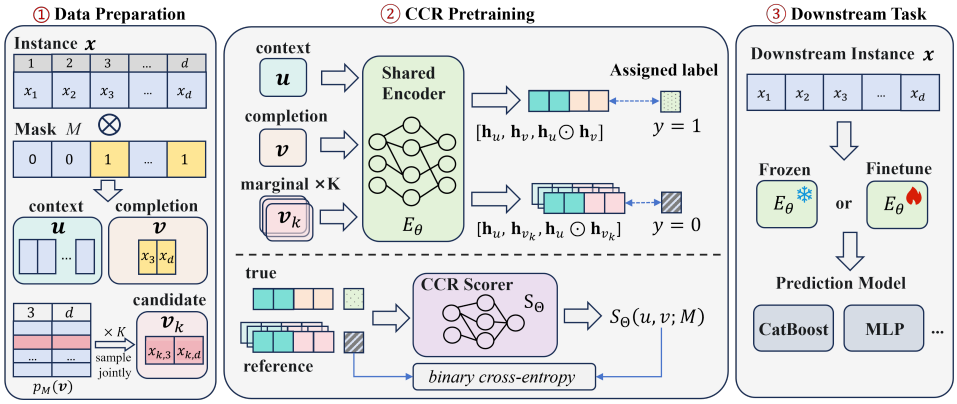


Figure 2: Overview of CCR. For each mask M, a sample is decomposed into context $\boldsymbol{u}$ and completion $\boldsymbol{v}$. CCR constructs a binary discrimination task between samples from the joint distribution $p_M(\boldsymbol{u}, \boldsymbol{v})$ and marginal-reference samples from $p_M(\boldsymbol{u})p_M(\boldsymbol{v})$. A shared encoder and a scoring network learn to estimate the conditional-to-marginal probability shift.

a candidate completion becomes more or less likely after observing the context. Mathematically, this shift corresponds exactly to the logarithmic density ratio between the conditional and marginal distributions.

4.1 The Computational Cost of Direct Conditional Modeling

A direct way to obtain the probability shift in Eq. (2) is to estimate $p_M(\boldsymbol{v} \mid \boldsymbol{u})$ and $p_M(\boldsymbol{v})$ separately and take their ratio. Autoregressive or mixture models can make this estimation tractable. However, this approach still requires learning two multivariate, mixed-type distributions. Errors in either estimate are inherited by the ratio and can be amplified where $p_M(\boldsymbol{v})$ is small (Sugiyama et al., 2010). For a joint model without a tractable factorization, evaluating the conditional distribution requires a context-dependent denominator over the completion space:

$$p_M(\boldsymbol{v} \mid \boldsymbol{u}) = \frac{p_M(\boldsymbol{u}, \boldsymbol{v})}{\int_{\mathcal{X}_M} p_M(\boldsymbol{u}, \boldsymbol{v}')\, d\mu_M(\boldsymbol{v}')}, \tag{3}$$

where the denominator involves integration over numerical variables and summation over categorical variables. Since $\mathcal{X}_M$ may combine both types of features, its exact computation becomes more difficult as the number and cardinality of the target columns grow.

Moreover, $p_M(\boldsymbol{v} \mid \boldsymbol{u})$ may be multimodal. A single context can admit several plausible completions, such as different occupations for the same age and education. Single-point reconstruction may average these modes, whereas representing them requires a richer conditional model. Since Eq. (2) only requires the conditional-to-marginal ratio, we estimate this ratio directly rather than first solving two separate density-estimation problems.

4.2 Estimating Conditional-Marginal Shift via CCR

To estimate the contextual probability shift without explicitly evaluating the normalized conditional distribution in Eq. (3), CCR reformulates the problem as a binary discrimination task.

Specifically, for a sampled mask M, an observed context $\boldsymbol{u}$ and its corresponding completion $\boldsymbol{v}$ are jointly drawn from $p_M(\boldsymbol{u}, \boldsymbol{v})$ and assigned $y = 1$. For the same context, we independently sample K candidate completions $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_K$ from the marginal completion distribution $p_M(\boldsymbol{v})$. Combining $\boldsymbol{u}$ with each independently sampled completion produces samples from $p_M(\boldsymbol{u})p_M(\boldsymbol{v})$, which are assigned $y = 0$. Here, $y = 0$ only indicates that the candidate completion is sampled from the marginal distribution independently of the current context. It does not imply semantic invalidity.

We implement this objective with a score function $S(\boldsymbol{u}, \boldsymbol{v}; M)$, which maps a context $\boldsymbol{u}$ and a candidate completion $\boldsymbol{v}$ under mask M to a scalar logit. CCR is optimized by binary cross-entropy:

$$\mathcal{L}_{\mathrm{CCR},M}(S) = \mathbb{E}\left[\log\left(1 + e^{-S(\boldsymbol{u},\boldsymbol{v};M)}\right) + \frac{1}{K}\sum_{k=1}^{K}\log\left(1 + e^{S(\boldsymbol{u},\boldsymbol{v}_k;M)}\right)\right] \tag{4}$$

The first term encourages high scores for instances with $y = 1$, while the second term encourages low scores for instances with $y = 0$.

Proposition 1 (CCR as Conditional-Marginal Shift Estimation)**.** *Fix an arbitrary mask M. Assume the score function S has sufficient capacity and that the objective in Eq.* (4) *is optimized over all measurable scores. Then, for almost every $(\boldsymbol{u}, \boldsymbol{v})$ for which the density ratio is well defined, the optimal score satisfies:*

$$S^\star(\boldsymbol{u}, \boldsymbol{v}; M) = \log \frac{p_M(\boldsymbol{v} \mid \boldsymbol{u})}{p_M(\boldsymbol{v})} = r_M(\boldsymbol{u}, \boldsymbol{v}) \tag{5}$$

Proof Sketch. Proposition 1 characterizes the optimal CCR score as the conditional-to-marginal log-density ratio under the binary discrimination framework (Gutmann and Hyvärinen, 2010). Under the equal weighting in Eq. (4), the Bayes-optimal posterior odds for distinguishing samples from the joint distribution $p_M(\boldsymbol{u}, \boldsymbol{v})$ and the product of marginals $p_M(\boldsymbol{u})p_M(\boldsymbol{v})$ are exactly $p_M(\boldsymbol{v} \mid \boldsymbol{u})/p_M(\boldsymbol{v})$. Since CCR estimates this ratio directly from samples of the two distributions, the context-dependent normalizer $\int_{\mathcal{X}_M} p_M(\boldsymbol{u}, \boldsymbol{v}')\, d\mu_M(\boldsymbol{v}')$ does not need to be evaluated explicitly. A complete proof is provided in Appendix A.1. We also show that the learned CCR score closely recovers the oracle density ratio across varying dependency strengths in Appendix D.4.

4.3 Neural Parameterization for CCR

How $\boldsymbol{u}$ and $\boldsymbol{v}$ are formed determines which dependencies the encoder is trained to capture. ICE-T contrasts a single column with the rest, while T-JEPA predicts target embeddings contextualized by the full row. CCR encodes $\boldsymbol{u}$ and $\boldsymbol{v}$ using only the observed values within each subset. Each candidate completion takes all target features from the same independently sampled training instance, preserving dependencies among the completion features. Both subsets use a shared encoder E_θ with lightweight projections:

$$\boldsymbol{h}_u = \mathrm{SiLU}\left(\boldsymbol{W}_u\, \mathrm{LN}(E_\theta(\boldsymbol{u}; M))\right), \qquad \boldsymbol{h}_v = \mathrm{SiLU}\left(\boldsymbol{W}_v\, \mathrm{LN}(E_\theta(\boldsymbol{v}; M))\right) \tag{6}$$

where $\mathrm{LN}(\cdot)$ is Layer Normalization and $\boldsymbol{W}_u, \boldsymbol{W}_v$ are learnable matrices. Resampling M trains the shared encoder across diverse column subsets in both context and completion. A two-layer MLP f_ω scores their compatibility using the projected representations and their elementwise product:

$$S_\Theta(\boldsymbol{u}, \boldsymbol{v}; M) = f_\omega\left([\boldsymbol{h}_u, \boldsymbol{h}_v, \boldsymbol{h}_u \odot \boldsymbol{h}_v]\right) \tag{7}$$

where $[\cdot]$ denotes concatenation and $\odot$ denotes elementwise multiplication. After pretraining, we discard the projection and scoring heads and use $E_\theta(\boldsymbol{x})$ as the downstream representation.

5 Experiments

To validate the effectiveness of CCR, we designed experiments to answer three key research questions: **RQ1:** Does CCR improve downstream few-shot classification? **RQ2:** Does CCR learn context-dependent completion support? **RQ3:** Which design choices contribute to the performance of CCR? Additional results and computational complexity analysis are provided in Appendix D.

Datasets. We evaluate CCR on 50 tabular classification datasets from the OpenML-CC18 (Bischl et al., 2017) and TabArena (Erickson et al., 2026) benchmarks, encompassing continuous, categorical, and mixed-type tables. Each dataset is partitioned into training/validation/test (70%/10%/20%) splits shared by all methods. The validation set is used for SSL early stopping. The metadata and selection criteria for the datasets are provided in Appendix B. Baselines follow their official preprocessing. For CCR, missing values are imputed using the training-set mean and mode, after which numerical features are standardized and categorical features are one-hot encoded. Masking is applied before one-hot encoding. Downstream labels are not used during SSL pretraining.

Evaluation Protocol. We use a frozen-representation evaluation protocol, while results under supervised fine-tuning are also reported. After pretraining, the encoder is frozen and used to obtain representations for the training and test instances. A classifier is then trained on the frozen representations using k labeled examples per class. We evaluate $k \in \{1, 5, 10\}$ and additionally report full-data results for reference. Unless otherwise stated, CatBoost (Prokhorenkova et al., 2018) is used as the downstream classifier because it provides strong and stable performance in tabular data.

Baselines. We compare CCR with eight representative tabular SSL methods: VIME (Yoon et al., 2020), SubTab (Ucar et al., 2021) and SwitchTab (Wu et al., 2024), which use reconstruction-based objectives; SCARF (Bahri et al., 2022), which uses contrastive view alignment; SAINT (Somepalli et al., 2021), which combines reconstruction and contrastive objectives; T-JEPA (Thimonier et al., 2025), which predicts latent representations; and ICE-T (Tokar and Sanner, 2025), which contrasts one column against the aggregation of other columns. We also include CAT (short for CatBoost (Prokhorenkova et al., 2018)), trained directly on the preprocessed input features, as a supervised baseline; and STUNT (Nam et al., 2023), which meta-learns from pseudo-tasks generated from unlabeled data. Comparison with tabular foundation models such as TabPFN (Hollmann et al., 2025) and TabICL (Qu et al., 2025) can be found in Appendix D.8.

Parameters. CCR uses a two-layer MLP encoder with a hidden dimension of 256 and an output dimension of 128. For each training instance, we sample 30% features as the target, with at least one feature and draw $K = 4$ marginal candidate completions. We train the model using Adam with a batch size of 256 and a learning rate of 10^{-3} for at most 300 epochs. Early stopping is applied with a patience of 100 based on the SSL validation loss, without using downstream labels. The hyperparameter settings and baselines are provided in Appendix C. Sensitivity analyses for the embedding dimension, the selection of K, and different evaluators are provided in Appendix D.2.

Metrics. We focus on two metrics, Accuracy and Macro-F1. Since they show similar trends, we mainly report accuracy in the main text and provide Macro-F1 in Appendix D.12. Main results are reported as mean $\pm$ standard deviation over 100 random seeds (42, 1042, . . ., 99042).

5.1 RQ1: Performance under Limited Labels

In this part, we evaluate whether CCR improves downstream classification when only limited labeled samples are available and analyze the effect of mask ratio.

Performance in Few-Shot Regimes. Table 1 reports the average accuracy ranks across 50 datasets under frozen-representation evaluation (Frz) and supervised fine-tuning (FT), with full rankings provided in Appendix D.11. In Frz, a downstream classifier is trained on fixed representations, whereas FT jointly updates the pretrained encoder and a 2-layer MLP head. Since CatBoost is trained directly on the original features, it has no corresponding FT result. CCR achieves the best average rank in all few-shot settings under both protocols. Under full supervision, CatBoost performs best in Frz, while CCR remains the best SSL method in FT. These results show that the advantage of CCR is most pronounced when labeled data are limited.

Method	1-shot Frz | FT	5-shot Frz | FT	10-shot Frz | FT	Full Frz | FT
CatBoost	7.1 | –	5.3 | –	4.8 | –	**1.7** | –
VIME	6.6 | 7.7	7.4 | 7.6	7.4 | 7.5	7.0 | 6.8
SCARF	4.4 | 4.5	5.9 | 5.0	5.8 | 4.8	5.5 | 4.5
T-JEPA	6.2 | 3.4	5.6 | 3.1	4.6 | 2.8	3.4 | 3.0
ICE-T	5.4 | 4.7	5.6 | 5.2	6.0 | 5.1	4.8 | 4.8
CCR	**2.2** | **1.9**	**2.1** | **1.9**	**1.8** | **2.1**	2.5 | **2.2**

Table 1: Average accuracy ranks (↓) across all methods and 50 datasets; representative methods are shown.

Effect of the mask ratio. Fig. 3 shows that CCR is relatively stable under moderate mask ratios. 1-shot evaluation performs best at 0.2, while 5-shot and 10-shot evaluation slightly favor 0.4–0.5; this may be because, under an extremely limited label budget, a less aggressive pretext task favors stable and readily accessible dependencies. Performance consistently declines when the mask ratio exceeds 0.5, suggesting that excessive masking removes too much context for estimating the conditional shift. Overall, the default ratio of 0.3 provides robust performance across label budgets.

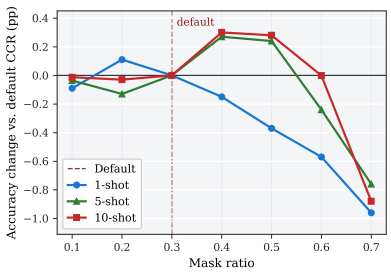


Figure 3: Accuracy change relative to the default ratio(0.3) across 50 datasets.

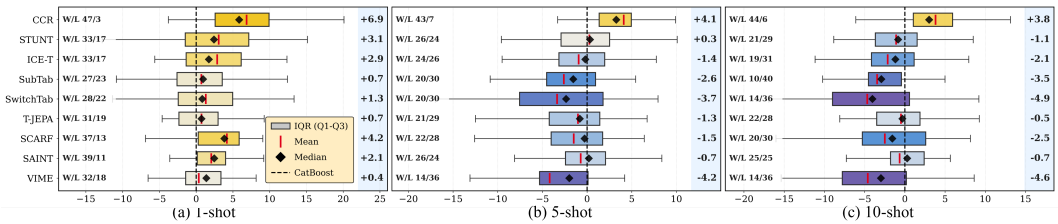
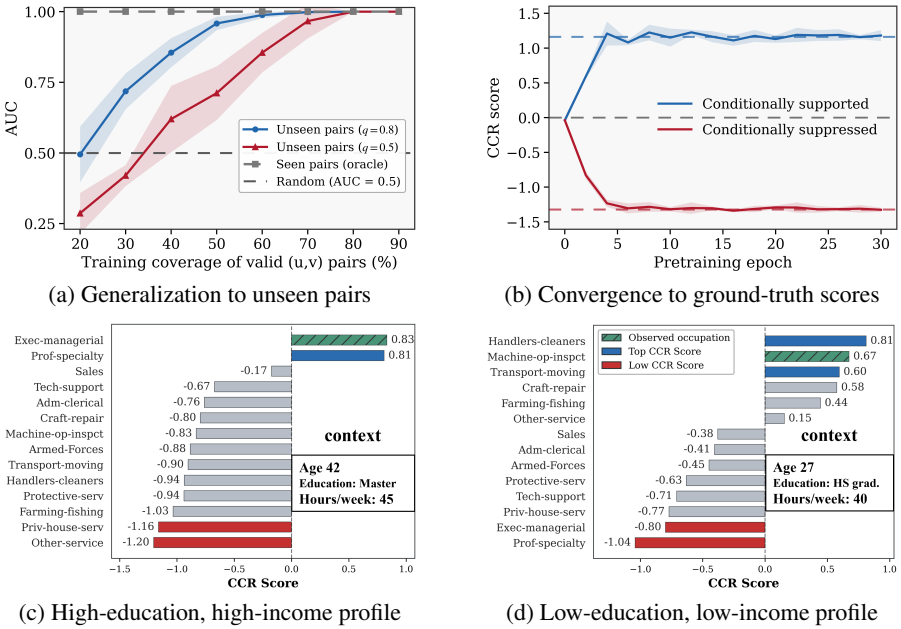


Figure 4: Dataset-level distributions of accuracy improvements over CatBoost under different few-shot settings. Boxes summarize the distribution across datasets, and the dashed vertical line indicates no improvement over CatBoost.

Fig. 4 summarizes the dataset-level accuracy improvement over the CatBoost baseline under different few-shot budgets. Overall, CCR achieves the most consistent positive improvement among all SSL methods. In the extremely low-label regime (1-shot), CCR obtains the highest median and mean improvement, and its interquartile range lies largely above zero. This indicates that the representations learned by CCR provide useful information even when the downstream classifier has access to only one labeled example per class.

5.2 Does CCR Recover the Conditional-to-Marginal Probability Shift?

To examine whether CCR generalizes beyond observed pairs, we construct 6,000 four-feature samples, with $\boldsymbol{u} = (x_1, x_2)$ and $\boldsymbol{v} = (x_3, x_4)$ as context and completion. A parameter q controls the strength of the dependency $x_1 + x_2 = x_3 + x_4$, with larger q indicating stronger context–completion dependence. For $q = 0.8$, the oracle scores are approximately 1.16 for supported pairs and -1.32 for suppressed pairs. We vary the coverage of supported training pairs at a fixed training size. Every held-out pair combines a context and completion seen individually during pretraining but never together. Details of the dataset construction are provided in Appendix B.2.



(a) Generalization to unseen pairs

(b) Convergence to ground-truth scores

(c) High-education, high-income profile

(d) Low-education, low-income profile

Figure 5: Analysis of CCR on synthetic and real-world data. (a) Compatibility AUC on observed and held-out $(\boldsymbol{u}, \boldsymbol{v})$ combinations as the coverage of supported training pairs increases. (b) CCR scores during pretraining. Blue denotes supported pairs ($x_1 + x_2 = x_3 + x_4$), while red denotes suppressed pairs ($x_1 + x_2 \neq x_3 + x_4$). (c–d) CCR compatibility scores for candidate occupations under two Adult profiles selected by income, which is used only for profile selection and is not provided to CCR.

Fig. 5a shows that, for $q = 0.8$, held-out AUC increases from approximately 0.72 at 30% coverage to nearly 1.0 at 60%. Under weaker dependence ($q = 0.5$), performance improves more

slowly, approaching 1.0 at 80% coverage. CCR thus generalizes to unseen combinations in both settings, with stronger dependence enabling generalization from fewer observed combinations. Fig. 5b further shows that supported and suppressed scores separate rapidly and approach their respective oracle values. These results support both generalization beyond observed pairs and recovery of the conditional-to-marginal probability shift. We also show that CCR learns useful representations that capture feature dependencies in Appendix D.5.

Analysis of Real-World Tabular Contexts. To illustrate CCR on real-world data, we visualize the learned compatibility scores for occupation under two Adult profiles selected from different income groups. Income is used only to select the profiles and is never provided to CCR during pretraining or scoring. As shown in Figs. 5c and 5d, CCR assigns positive scores to `Exec-managerial` and `Prof-specialty` for the higher-income profile with advanced education, but favors blue-collar occupations for the lower-income profile. More fundamentally, the CCR score quantifies the context-specific importance of each candidate feature value: its sign identifies whether the context promotes or suppresses that value relative to its marginal frequency, while its magnitude measures the strength of this shift. We further discuss this in Appendix A.1.

5.3 RQ3: What Drives the Effectiveness of CCR?

To examine the design choices within CCR, we compare CCR with three masking variants: (i) *CCR fixed mask*: at the beginning of each run, a subset of target columns is sampled from the same mask distribution used by CCR and kept fixed throughout pretraining; (ii) *CCR single-feature mask*: randomly selects one feature at each training step; and (iii) *CCR Pearson mask*: selects features with higher Pearson scores. Additionally, we consider a scorer ablation: (iv) *CCR w/o Interaction*: the target columns are resampled during pretraining as in CCR, but the multiplicative term $\boldsymbol{h}_u \odot \boldsymbol{h}_v$ is removed from the scorer.

Table 2: Ablation of the masking strategy and interaction term in CCR. Results are average downstream accuracy rank(↓) against baselines across 50 datasets, lower is better.

Variant	Sampling	Product Term	1-shot	5-shot	10-shot	Full data
CCR fixed mask	Fixed	✓	2.93	2.94	2.91	3.14
CCR single-feature mask	One col.	✓	2.74	2.86	2.55	3.01
CCR Pearson mask	Correlation	✓	2.91	3.05	2.98	3.42
CCR w/o Interaction	Random	×	2.42	2.63	2.57	3.07
CCR	Random	✓	**2.23**	**2.09**	**1.83**	**2.52**

Table 2 shows that each design choice contributes to CCR. The fixed mask restricts the model to the same target features throughout pretraining, which limits the range of feature dependencies it can learn. The single-feature mask also performs worse than the default strategy, suggesting that jointly masking multiple features provides richer dependency signals. Pearson-guided masking underperforms random masking as well. Although it gives priority to strongly correlated features, it may place too much emphasis on a small set of pairwise linear relationships and provide less coverage of other feature dependencies. Adding the product term further improves downstream rank by making context–completion interactions directly available to the scorer.

6 Conclusion

In this work, we revisit tabular self-supervised learning from the perspective of conditional dependency modeling and propose Conditional Completion Ranking (CCR). By learning the conditional-to-marginal shift between observed and marginally sampled completions, CCR captures feature interactions without explicitly estimating complex conditional distributions. Across 50 tabular classification datasets, CCR achieves the best average rank among the evaluated methods in few-shot settings, while further analyses support its ability to recover conditional compatibility structures. These results highlight the potential of conditional dependency modeling as an effective alternative for tabular representation learning. Future work will explore shared pretraining across heterogeneous tables to transfer learned dependencies across datasets.

REFERENCES

Sercan Ö Arik and Tomas Pfister. Tabnet: Attentive interpretable tabular learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 6679–6687, 2021.

Dara Bahri, Heinrich Jiang, Yi Tay, and Donald Metzler. Scarf: Self-supervised contrastive learning using random feature corruption. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=CuV_qYkmKb3.

Bernd Bischl, Giuseppe Casalicchio, Matthias Feurer, Pieter Gijsbers, Frank Hutter, Michel Lang, Rafael G Mantovani, Jan N van Rijn, and Joaquin Vanschoren. Openml benchmarking suites. *arXiv preprint arXiv:1708.03731*, 2017.

Jinglin Chen and Nan Jiang. Offline reinforcement learning under value and density-ratio realizability: the power of gaps. In *Uncertainty in Artificial Intelligence*, pages 378–388. PMLR, 2022.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

Kristy Choi, Chenlin Meng, Yang Song, and Stefano Ermon. Density ratio estimation via infinitesimal classification. In *International Conference on Artificial Intelligence and Statistics*, pages 2552–2573. PMLR, 2022.

Thomas M Cover and Joy A Thomas. Elements of information theory, 2/e. 2006.

Arnaud Delaunoy, Joeri Hermans, François Rozet, Antoine Wehenkel, and Gilles Louppe. Towards reliable simulation-based inference with balanced neural ratio estimation. *Advances in Neural Information Processing Systems*, 35:20025–20037, 2022.

Nick Erickson, Lennart Purucker, Andrej Tschalzev, David Holzmüller, Prateek Desai, David Salinas, and Frank Hutter. Tabarena: A living benchmark for machine learning on tabular data. *Advances in Neural Information Processing Systems*, 38, 2026.

Jie Gui, Tuo Chen, Jing Zhang, Qiong Cao, Zhenan Sun, Hao Luo, and Dacheng Tao. A survey on self-supervised learning: Algorithms, applications, and future trends. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):9052–9071, 2024.

Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 297–304. JMLR Workshop and Conference Proceedings, 2010.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pages 15979–15988. IEEE, 2022.

R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, 2018.

Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 637(8045):319–326, 2025.

Vasily Ilin, Peter Sushko, and Ranjay Krishna. Discoformer: Plug-in density and score estimation with transformers. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=gyOWJpP8cQ.

Changhee Lee, Fergus Imrie, and Mihaela van der Schaar. Self-supervision enhanced feature selection with correlated gates. In *International conference on learning representations*, 2022.

Kyungeun Lee, Moonjung Eo, Hye-Seung Cho, Min-Kook Suh, Seoyoon Kim, Ye Seul Sim, Suhee Yoon, Sanghyu Yoon, and Woohyung Lim. Range-limited augmentation for few-shot learning in tabular data with comprehensive benchmark. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 1262–1273, 2025.

Jaehyun Nam, Jihoon Tack, Kyungmin Lee, Hankook Lee, and Jinwoo Shin. Stunt: Few-shot tabular learning with self-generated tasks from unlabeled tables. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=_xlsjehDvlY.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. Catboost: unbiased boosting with categorical features. *Advances in neural information processing systems*, 31, 2018.

Jingang Qu, David Holzmüller, Gaël Varoquaux, and Marine Le Morvan. Tabicl: A tabular foundation model for in-context learning on large data. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 50817–50847. PMLR, 2025.

Benjamin Rhodes, Kai Xu, and Michael Gutmann. Telescoping density-ratio estimation. *Advances in neural information processing systems*, 33:4905–4916, 2020.

Claude Elwood Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

Gowthami Somepalli, Micah Goldblum, Avi Schwarzschild, C Bayan Bruss, and Tom Goldstein. Saint: Improved neural networks for tabular data via row attention and contrastive pre-training. *arXiv preprint arXiv:2106.01342*, 2021.

Masashi Sugiyama, Shinichi Nakajima, Hisashi Kashima, Paul Buenau, and Motoaki Kawanabe. Direct importance estimation with model selection and its application to covariate shift adaptation. *Advances in neural information processing systems*, 20, 2007.

Masashi Sugiyama, Ichiro Takeuchi, Taiji Suzuki, Takafumi Kanamori, Hirotaka Hachiya, and Daisuke Okanohara. Conditional density estimation via least-squares density ratio estimation. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 781–788. JMLR Workshop and Conference Proceedings, 2010.

Masashi Sugiyama, Taiji Suzuki, and Takafumi Kanamori. *Density ratio estimation in machine learning*, volume 10. Cambridge University Press Cambridge, 2012.

Taiji Suzuki, Masashi Sugiyama, Jun Sese, and Takafumi Kanamori. Approximating mutual information by maximum likelihood density ratio estimation. In *New challenges for feature selection in data mining and knowledge discovery*, pages 5–20. PMLR, 2008.

Rahul Thapa, Bryan He, Magnus Ruud Kjaer, Hyatt Moore Iv, Gauri Ganjoo, Emmanuel Mignot, and James Zou. SleepFM: Multi-modal representation learning for sleep across brain activity, ECG and respiratory signals. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pages 48019–48037. PMLR, 2024.

Hugo Thimonier, José Lucas de Melo Costa, Fabrice Popineau, Arpad Rimmel, and Bich-Liên Doan. T-jepa: Augmentation-free self-supervised learning for tabular data. In *International Conference on Learning Representations*, volume 2025, pages 58743–58768, 2025.

Tomas Tokar and Scott Sanner. Ice-t: Interactions-aware cross-column contrastive embedding for heterogeneous tabular datasets. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 20904–20911, 2025.

Christopher Tosh, Akshay Krishnamurthy, and Daniel Hsu. Contrastive learning, multi-view redundancy, and linear models. In *Algorithmic learning theory*, pages 1179–1206. PMLR, 2021.

Talip Ucar, Ehsan Hajiramezanali, and Lindsay Edwards. Subtab: Subsetting features of tabular data for self-supervised representation learning. *Advances in Neural Information Processing Systems*, 34:18853–18865, 2021.

Meilin Wang, Wei Huang, Mingming Gong, and Zheng Zhang. Projection pursuit density ratio estimation. In *Forty-second International Conference on Machine Learning*, 2025a. URL https://openreview.net/forum?id=MgNeJO0PcF.

Wei-Yao Wang, Wei-Wei Du, Derek Xu, Wei Wang, and Wen-Chih Peng. A survey on self-supervised learning for non-sequential tabular data. *Machine Learning*, 114(1):16, 2025b.

Zifeng Wang and Jimeng Sun. Transtab: Learning transferable tabular transformers across tables. *Advances in Neural Information Processing Systems*, 35:2902–2915, 2022.

Ladyna Wittscher. A survey on design choices for self-supervised learning in computer vision. *Artificial Intelligence Review*, 59(4):112, 2026.

Jing Wu, Suiyao Chen, Qi Zhao, Renat Sergazinov, Chen Li, Shengjie Liu, Chongchao Zhao, Tianpei Xie, Hanqing Guo, Cheng Ji, et al. Switchtab: Switched autoencoders are effective tabular learners. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 15924–15933, 2024.

Shuntuo Xu, Zhou Yu, and Jian Huang. Estimating unbounded density ratios: Applications in error control under covariate shift. *arXiv preprint arXiv:2504.01031*, 2025.

Jinsung Yoon, Yao Zhang, James Jordon, and Mihaela Van der Schaar. Vime: Extending the success of self-and semi-supervised learning to tabular domain. *Advances in neural information processing systems*, 33:11033–11043, 2020.

Jianping Zhang, Lei Yang, Seyed Mahmoud Sajjadi Mohammadabadi, and Feng Yan. A survey on self-supervised learning: Recent advances and open problems. *Neurocomputing*, page 131409, 2025.

Bingzhao Zhu, Xingjian Shi, Nick Erickson, Mu Li, George Karypis, and Mahsa Shoaran. Xtab: Cross-table pretraining for tabular transformers. *Proceedings of Machine Learning Research*, 202:43181–43204, 2023.

APPENDIX CONTENTS

Usage Statement of LLM **14**

A Additional Details of CCR **14**

A.1 Proof of Proposition 1 14

A.2 CCR-Cal: Weight Calibration with Limited Unlabeled Data 15

A.3 Proof of Proposition 2 16

A.4 Parameter Setting of CCR 17

A.5 Pseudocode 17

B Dataset Description **17**

B.1 Real-world Datasets 17

B.2 Synthetic Dataset Construction 19

C Details of Implementing Baselines **19**

D Additional Results **21**

D.1 Significance Test 21

D.2 Hyperparameter Sensitivity and Evaluator Robustness 22

D.3 Does pretraining help beyond the encoder architecture? 23

D.4 Verification of Density-Ratio Recovery 24

D.5 Does the encoder preserve the learned feature dependencies? 24

D.6 Comparing Pretraining Objectives with a Fixed Backbone 25

D.7 Analysis of CCR-Cal with Limited Unlabeled Data 26

D.8 Comparison with Tabular Foundation Models 27

D.9 Computational Complexity 28

D.10 Limitations and Future Work 28

D.11 Average Rankings 29

D.12 Detailed Results 29

USAGE STATEMENT OF LLM

In this work, we used LLMs only to edit the manuscript for grammar correction and to improve overall readability. We have not used generative AI tools for any core research tasks. We have carefully reviewed all AI-assisted language edits to ensure that the original scientific meaning and accuracy remain strictly intact.

A ADDITIONAL DETAILS OF CCR

A.1 PROOF OF PROPOSITION 1

We prove the density-ratio characterization of the CCR objective. Fix a target subset M and a context value $\boldsymbol{u}$. For compactness, define:

$$p_d(\boldsymbol{v}) = p_M(\boldsymbol{v} \mid \boldsymbol{u}), \qquad p_n(\boldsymbol{v}) = p_M(\boldsymbol{v})$$

Here p_d denotes the conditional completion distribution under the current context, while p_n denotes the marginal completion distribution. For mixed continuous and discrete features, all densities or probability masses are understood with respect to the same product reference measure on $\mathcal{X}_M$.

Let $q(\boldsymbol{v}) = \sigma\left(S(\boldsymbol{u}, \boldsymbol{v}; M)\right)$ denote the probability that $(\boldsymbol{u}, \boldsymbol{v})$ comes from the joint distribution rather than the product of marginals, where $\sigma(t) = (1 + \exp(-t))^{-1}$ denotes the logistic sigmoid function. For the fixed context $\boldsymbol{u}$, the CCR objective in Eq. (4), with the expectation taken over the underlying data distribution, can be written as:

$$\mathcal{J}_{\boldsymbol{u}}(q) = -\int_{\mathcal{X}_M} \left[p_d(\boldsymbol{v}) \log q(\boldsymbol{v}) + p_n(\boldsymbol{v}) \log\left(1 - q(\boldsymbol{v})\right)\right] d\mu_M(\boldsymbol{v}) \tag{8}$$

Since the objective is pointwise separable over $\boldsymbol{v}$, its minimizer can be obtained by minimizing:

$$\ell_{\boldsymbol{v}}(q) = -p_d(\boldsymbol{v}) \log q - p_n(\boldsymbol{v}) \log(1 - q) \tag{9}$$

for each $\boldsymbol{v}$ independently. The first-order condition is:

$$-\frac{p_d(\boldsymbol{v})}{q} + \frac{p_n(\boldsymbol{v})}{1 - q} = 0 \tag{10}$$

which yields:

$$q^*(\boldsymbol{v}) = \frac{p_d(\boldsymbol{v})}{p_d(\boldsymbol{v}) + p_n(\boldsymbol{v})} \tag{11}$$

Moreover,

$$\frac{d^2 \ell_{\boldsymbol{v}}(q)}{dq^2} = \frac{p_d(\boldsymbol{v})}{q^2} + \frac{p_n(\boldsymbol{v})}{(1 - q)^2} > 0 \tag{12}$$

so $q^*(\boldsymbol{v})$ is the global minimizer whenever the density ratio is well defined.

Taking the logit of $q^*(\boldsymbol{v})$ gives:

$$S^*(\boldsymbol{u}, \boldsymbol{v}; M) = \log \frac{q^*(\boldsymbol{v})}{1 - q^*(\boldsymbol{v})} \tag{13}$$

$$= \log \frac{p_d(\boldsymbol{v})}{p_n(\boldsymbol{v})} \tag{14}$$

$$= \log \frac{p_M(\boldsymbol{v} \mid \boldsymbol{u})}{p_M(\boldsymbol{v})} = r_M(\boldsymbol{u}, \boldsymbol{v}) \tag{15}$$

Thus, the optimal CCR score recovers the conditional-to-marginal log-density ratio. □

Context-Specific Feature-Value Attribution. Proposition 1 also gives the CCR score a direct value-level interpretation. When M contains a single categorical feature, each candidate $\boldsymbol{v}$ corresponds to one possible value of that feature, and

$$S^*(\boldsymbol{u}, \boldsymbol{v}; M) = \log \frac{p_M(\boldsymbol{v} \mid \boldsymbol{u})}{p_M(\boldsymbol{v})}$$

Therefore, the sign of the score indicates whether the context promotes or suppresses the candidate relative to its marginal frequency, while its magnitude quantifies the strength of this change on a logarithmic scale. Equivalently,

$$\exp(S^*(\boldsymbol{u}, \boldsymbol{v}; M)) = \frac{p_M(\boldsymbol{v} \mid \boldsymbol{u})}{p_M(\boldsymbol{v})}$$

which gives the multiplicative probability shift induced by the context. Thus, for a fixed context, CCR provides a context-specific attribution over the candidate values of the target feature, rather than a global importance score for the feature itself.

A.2 CCR-CAL: WEIGHT CALIBRATION WITH LIMITED UNLABELED DATA

Proposition 1 shows that the optimal CCR score recovers the conditional-to-marginal log-density ratio. When the training data are limited, however, a completion sampled from another row may still be plausible under the current context. For example, two individuals with similar age and education may have different occupations, both of which remain plausible under either context. Nevertheless, when one occupation is sampled as a marginal candidate for the other individual, the CCR objective treats this completion as a marginal candidate and encourages the model to lower its score. This motivates exploring whether downweighting such candidates improves representation learning with limited unlabeled data.

For each marginal candidate $\boldsymbol{v}_k$, let $\boldsymbol{u}_k$ denote the context from the same sampled row. We use the similarity between the current context $\boldsymbol{u}$ and $\boldsymbol{u}_k$ to adjust the contribution of $\boldsymbol{v}_k$ to the loss:

$$w_k = f\left(\frac{\boldsymbol{h}_{\boldsymbol{u}}^\top \boldsymbol{h}_{\boldsymbol{u}_k}}{\|\boldsymbol{h}_{\boldsymbol{u}}\|_2 \|\boldsymbol{h}_{\boldsymbol{u}_k}\|_2}\right), \qquad w_k \in [w_{\min}, 1], \tag{16}$$

where $\boldsymbol{h}_{\boldsymbol{u}}$ and $\boldsymbol{h}_{\boldsymbol{u}_k}$ are the representations of the two contexts, and f decreases as their similarity increases. Thus, a candidate sampled from a more similar context receives a smaller weight. The calibrated objective is:

$$\mathcal{L}_{\text{cal}}(S_\Theta) = \mathbb{E}\left[\log\left(1 + e^{-S_\Theta(\boldsymbol{u},\boldsymbol{v};M)}\right) + \frac{1}{K}\sum_{k=1}^{K} w_k \log\left(1 + e^{S_\Theta(\boldsymbol{u},\boldsymbol{v}_k;M)}\right)\right] \tag{17}$$

When $w_k = 1$ for all candidates, Eq. (17) reduces to the unweighted CCR objective. To see how this sample-level weighting changes the optimum, we consider the average weight assigned to a given context and candidate. Let $\bar{w}_M(\boldsymbol{u}, \boldsymbol{v}) = \mathbb{E}[w_k \mid \boldsymbol{u}, \boldsymbol{v}_k = \boldsymbol{v}]$ denote the average weight assigned to candidate $\boldsymbol{v}$ under context $\boldsymbol{u}$. For the following analysis, we view the weights as fixed during score optimization, consistent with stopping gradients through the weights during each update.

Proposition 2 (Bounded Calibration Effect under Fixed Weights)**.** *Fix an arbitrary mask M. Assuming the weighting function $\bar{w}_M(u, v)$ is exogenous or held constant with respect to the optimization of S, the population-optimal calibrated score satisfies:*

$$0 \le S^*_{\text{cal}}(\boldsymbol{u}, \boldsymbol{v}; M) - r_M(\boldsymbol{u}, \boldsymbol{v}) = -\log \bar{w}_M(\boldsymbol{u}, \boldsymbol{v}) \le \log \frac{1}{w_{\min}} \tag{18}$$

Under the same context, any pairwise ordering with a clean-score margin greater than $\log(1/w_{\min})$ is preserved after calibration.

Proposition 2 explains how calibration changes the CCR score when the weighting function is fixed. Since $\bar{w}_M(\boldsymbol{u}, \boldsymbol{v}) \le 1$, calibration can only raise the score of a marginal candidate, consistent with our motivation to reduce the penalty on plausible alternative completions. The increase is bounded by $\log(1/w_{\min})$, so calibration cannot move the score arbitrarily far from the unweighted CCR optimum. Only candidates whose unweighted scores differ by at most this amount can be reordered. Thus, $w_{\min}$ directly controls the strength of calibration: a larger $w_{\min}$ gives a smaller adjustment, while $w_{\min} = 1$ removes the adjustment entirely. This calibrated objective is treated as an optional variant; the main CCR method and its population characterization remain those defined in Eq. (4) and Proposition 1.

A.3 Proof of Proposition 2

Fix a target subset M and a context value $\boldsymbol{u}$. Following the proof of Proposition 1, we define:

$$p_d(\boldsymbol{v}) = p_M(\boldsymbol{v} \mid \boldsymbol{u}), \qquad p_n(\boldsymbol{v}) = p_M(\boldsymbol{v}),$$

and let $q(\boldsymbol{v}) = \sigma\left(S_{\text{cal}}(\boldsymbol{u}, \boldsymbol{v}; M)\right)$, where $\sigma(t) = (1 + \exp(-t))^{-1}$ is the logistic sigmoid function. We treat the reliability weights as fixed during score optimization, consistent with stopping gradients. Recall that $\bar{w}_M(\boldsymbol{u}, \boldsymbol{v}) = \mathbb{E}\left[w_k \mid \boldsymbol{u}, \boldsymbol{v}_k = \boldsymbol{v}\right]$ is the expected weight assigned to a marginal candidate $\boldsymbol{v}$ under context $\boldsymbol{u}$. Taking the expectation over the sampled marginal candidates, the calibrated objective in Eq. (17) can be written as:

$$\mathcal{J}^{\text{cal}}_{\boldsymbol{u}}(q) = -\int_{\mathcal{X}_M} \left[p_d(\boldsymbol{v}) \log q(\boldsymbol{v}) + \bar{w}_M(\boldsymbol{u}, \boldsymbol{v}) p_n(\boldsymbol{v}) \log\left(1 - q(\boldsymbol{v})\right)\right] d\mu_M(\boldsymbol{v}) \tag{19}$$

Since the objective is pointwise separable over $\boldsymbol{v}$, its minimizer can be obtained by minimizing:

$$\ell^{\text{cal}}_{\boldsymbol{v}}(q) = -p_d(\boldsymbol{v}) \log q - \bar{w}_M(\boldsymbol{u}, \boldsymbol{v}) p_n(\boldsymbol{v}) \log(1 - q) \tag{20}$$

for each $\boldsymbol{v}$ independently. The first-order condition is:

$$-\frac{p_d(\boldsymbol{v})}{q} + \frac{\bar{w}_M(\boldsymbol{u}, \boldsymbol{v}) p_n(\boldsymbol{v})}{1 - q} = 0 \tag{21}$$

which yields:

$$q^*(\boldsymbol{v}) = \frac{p_d(\boldsymbol{v})}{p_d(\boldsymbol{v}) + \bar{w}_M(\boldsymbol{u}, \boldsymbol{v}) p_n(\boldsymbol{v})} \tag{22}$$

The pointwise objective is convex because:

$$\frac{d^2 \ell^{\text{cal}}_{\boldsymbol{v}}(q)}{dq^2} = \frac{p_d(\boldsymbol{v})}{q^2} + \frac{\bar{w}_M(\boldsymbol{u}, \boldsymbol{v}) p_n(\boldsymbol{v})}{(1 - q)^2} > 0 \tag{23}$$

Therefore, $q^*(\boldsymbol{v})$ is the global minimizer whenever the density ratio is well defined. Taking the logit of $q^*(\boldsymbol{v})$ gives:

$$S^*_{\text{cal}}(\boldsymbol{u}, \boldsymbol{v}; M) = \log \frac{p_d(\boldsymbol{v})}{\bar{w}_M(\boldsymbol{u}, \boldsymbol{v}) p_n(\boldsymbol{v})} \tag{24}$$

$$= r_M(\boldsymbol{u}, \boldsymbol{v}) - \log \bar{w}_M(\boldsymbol{u}, \boldsymbol{v}) \tag{25}$$

Since $w_k \in [w_{\min}, 1]$, its conditional expectation also satisfies:

$$w_{\min} \le \bar{w}_M(\boldsymbol{u}, \boldsymbol{v}) \le 1$$

It follows that:

$$0 \le S^*_{\text{cal}}(\boldsymbol{u}, \boldsymbol{v}; M) - r_M(\boldsymbol{u}, \boldsymbol{v}) = -\log \bar{w}_M(\boldsymbol{u}, \boldsymbol{v}) \le \log \frac{1}{w_{\min}} \tag{26}$$

Finally, consider two candidate completions $\boldsymbol{v}_a$ and $\boldsymbol{v}_b$ under the same context and mask. Define:

$$\delta(\boldsymbol{v}) = -\log \bar{w}_M(\boldsymbol{u}, \boldsymbol{v}), \qquad 0 \le \delta(\boldsymbol{v}) \le \log \frac{1}{w_{\min}} \tag{27}$$

Their calibrated score difference is:

$$\begin{aligned} &S^*_{\text{cal}}(\boldsymbol{u}, \boldsymbol{v}_a; M) - S^*_{\text{cal}}(\boldsymbol{u}, \boldsymbol{v}_b; M) \\ &= r_M(\boldsymbol{u}, \boldsymbol{v}_a) - r_M(\boldsymbol{u}, \boldsymbol{v}_b) + \delta(\boldsymbol{v}_a) - \delta(\boldsymbol{v}_b) \\ &\ge r_M(\boldsymbol{u}, \boldsymbol{v}_a) - r_M(\boldsymbol{u}, \boldsymbol{v}_b) - \log \frac{1}{w_{\min}} \end{aligned} \tag{28}$$

Therefore, if:

$$r_M(\boldsymbol{u}, \boldsymbol{v}_a) - r_M(\boldsymbol{u}, \boldsymbol{v}_b) > \log \frac{1}{w_{\min}}$$

then:

$$S^*_{\text{cal}}(\boldsymbol{u}, \boldsymbol{v}_a; M) > S^*_{\text{cal}}(\boldsymbol{u}, \boldsymbol{v}_b; M)$$

Hence, for fixed weights, calibration changes the population-optimal CCR score by a bounded amount and preserves any pairwise ordering whose unweighted-score margin exceeds this bound. □

A.4 Parameter Setting of CCR

Implementation of CCR. Let $\mathcal{D}_{\text{tr}} = \{\boldsymbol{x}_n\}_{n=1}^{N}$ denote the unlabeled training pool. For a training instance $\boldsymbol{x}_i$ and a sampled target subset M, we construct $\boldsymbol{u}_i = \boldsymbol{x}_{i,\bar{M}}$ and $\boldsymbol{v}_i^{+} = \boldsymbol{x}_{i,M}$. For each context $\boldsymbol{u}_i$, we independently sample K indices uniformly with replacement from the training pool:

$$J_{ik} \sim \text{Unif}\{1, \ldots, N\}, \qquad \boldsymbol{v}_{ik}^{-} = \boldsymbol{x}_{J_{ik},M}, \qquad k = 1, \ldots, K.$$

All target features in each $\boldsymbol{v}_{ik}^{-}$ come from the same training instance. Thus, candidates follow the empirical joint marginal distribution of the target features rather than the product of their individual marginals.

Construction of Masking M. Masking is performed at the original-column level. All one-hot dimensions associated with the same categorical column are assigned together. Let $\boldsymbol{m}_M$ denote the expanded binary mask corresponding to the target-column subset M. The context and positive-completion inputs are:

$$\boldsymbol{x}_{i,\text{ctx}} = \boldsymbol{x}_i \odot (1 - \boldsymbol{m}_M), \qquad \boldsymbol{c}_i^{+} = \boldsymbol{x}_i \odot \boldsymbol{m}_M \tag{29}$$

A target subset $M_i$is independently sampled for each training instance. For each context i and candidate k, a donor index J_{ik} is sampled with replacement from the SSL training pool, and the marginal completion is constructed as

$$\boldsymbol{c}_{ik}^{-} = \boldsymbol{x}_{J_{ik}} \odot \boldsymbol{m}_M \tag{30}$$

All target columns in $\boldsymbol{c}_{ik}^{-}$ therefore come from the same donor row. The mask projection is added to both views, enabling the encoder to distinguish an omitted column from an observed value equal to zero.

Optional calibration. For CCR-Cal, we compute the cosine similarity between the representations of the current context $\boldsymbol{u}$ and the donor context $\boldsymbol{u}_k$:

$$c_k = \frac{\boldsymbol{h}_{\boldsymbol{u}}^{\top} \boldsymbol{h}_{\boldsymbol{u}_k}}{\|\boldsymbol{h}_{\boldsymbol{u}}\|_2 \|\boldsymbol{h}_{\boldsymbol{u}_k}\|_2}. \tag{31}$$

The reliability weight is defined as

$$w_k = \max\left\{w_{\min}, 1 - [\max(c_k, 0)]^2\right\}, \qquad w_{\min} = 0.5. \tag{32}$$

Thus, negatively correlated or orthogonal contexts retain unit weight, whereas candidates drawn from more similar contexts are progressively downweighted. The lower bound limits the maximum score adjustment to $\log 2$. Gradients are stopped through w_k, and all other training settings are identical to CCR.

A.5 Pseudocode

Algorithm 1 summarizes the CCR pretraining procedure. For each context–completion pair $(\boldsymbol{u}, \boldsymbol{v})$, CCR samples K marginal completions from the full training set and optimizes either the objective or its reliability-calibrated variant. After pretraining, only the shared encoder $E_{\theta^\star}$ is retained for downstream tasks.

B Dataset Description

B.1 Real-world Datasets

We evaluate all methods on a diverse collection of classification datasets from two sources: OpenML-CC18 (Bischl et al., 2017) and TabArena (Erickson et al., 2026). The benchmark covers both classical tabular datasets and more recent large-scale tabular tasks, with substantial variation in sample size, feature dimensionality, number of classes, and feature types. The datasets include numerical-only, categorical-only, and mixed-type tables, allowing us to evaluate whether each representation learning method is robust across different forms of tabular heterogeneity.

Algorithm 1 CCR and Reliability-Calibrated CCR(CCR-Cal) Pretraining

Input: Unlabeled training data $\mathcal{D}_u$, mask distribution $q(M)$, and number of marginal candidates K
Output: Pretrained encoder $E_{\theta^\star}$

Initialize encoder, projection, and scoring parameters Θ
while not converged **do**
  Sample a target subset $M \sim q(M)$
  Sample an observed row from $\mathcal{D}_u$ and decompose it into context $\boldsymbol{u}$ and completion $\boldsymbol{v}$
  Independently sample K rows from the full training set $\mathcal{D}_u$
  Decompose them into context–completion pairs $\{(\boldsymbol{u}_k, \boldsymbol{v}_k)\}_{k=1}^{K}$ under the same mask M
  Compute $S_\Theta(\boldsymbol{u}, \boldsymbol{v}; M)$ and $\{S_\Theta(\boldsymbol{u}, \boldsymbol{v}_k; M)\}_{k=1}^{K}$ using Eq. (7)
  CCR
  Compute the objective using Eq. (4)
  CCR-Cal
  Alternatively, compute w_k from $\boldsymbol{u}$ and $\boldsymbol{u}_k$ using Eq. (16)
  Stop gradients through w_k and compute the objective using Eq. (17)
  Update Θ using the selected objective
end while
return $E_{\theta^\star}$

Table 3: Statistics of the OpenML datasets. *Samples*, *Features*, *Num.*, and *Cat.* denote the number of samples, total number of input features, numerical features, and categorical features, respectively.

Dataset ID	Classes	Samples	Features	Num.	Cat.	Name
6	26	20000	16	16	0	letter
12	10	2000	216	216	0	mfeat-factors
15	2	699	9	9	0	breast-w
16	10	2000	64	64	0	mfeat-karhunen
22	10	2000	47	47	0	mfeat-zernike
28	10	5620	64	64	0	optdigits
29	2	690	15	6	9	credit-approval
32	10	10992	16	16	0	pendigits
37	2	768	8	8	0	diabetes
38	2	3772	29	6	23	sick
44	2	4601	57	57	0	spambase
46	3	3190	60	0	60	splice
50	2	958	9	0	9	tic-tac-toe
182	6	6430	36	36	0	satimage
458	4	841	70	70	0	analcatdata_authorship
1067	2	2109	21	21	0	kc1
1068	2	1109	21	21	0	pc1
1468	9	1080	856	856	0	cnae-9
1478	6	10299	561	561	0	har
1480	2	583	10	9	1	ilpd
1485	2	2600	500	500	0	madelon
1489	2	5404	5	5	0	phoneme
1494	2	1055	41	41	0	qsar-biodeg
1501	10	1593	256	256	0	semeion
4534	2	11055	30	0	30	PhishingWebsites
4538	5	9873	32	32	0	GesturePhaseSegmentationProcessed
6332	2	540	37	18	19	cylinder-bands
23381	2	500	12	1	11	dresses-sales
40499	11	5500	40	40	0	texture
40668	3	67557	42	0	42	connect-4
40670	3	3186	180	0	180	dna
40979	10	2000	240	240	0	mfeat-pixel
40983	2	4839	5	5	0	wilt
40984	7	2310	16	16	0	segment

Tables 3 and 4 summarize the dataset statistics. For each dataset, we report the dataset identifier, number of classes, data shape, total number of input features, and the split between numerical and categorical features. Their sizes range from small few-hundred-sample datasets to large-scale

Table 4: Statistics of the TabArena datasets. *Samples*, *Features*, *Num.*, and *Cat.* denote the number of samples, total number of input features, numerical features, and categorical features, respectively.

Dataset ID	Classes	Samples	Features	Num.	Cat.	Name
46908	2	76000	170	170	0	APSFailure
46911	2	10000	10	6	4	Bank_Customer_Churn
46920	2	129880	21	5	16	customer_satisfaction_in_airline
46924	2	10999	10	6	4	E-CommereShippingData
46927	2	1500	6	3	3	Fitness_Club
46929	2	150000	10	10	0	GiveMeSomeCredit
46932	2	10459	23	23	0	heloc
46937	2	12684	24	2	22	in_vehicle_coupon_recommendation
46938	2	1723	13	5	8	Is-this-a-good-customer
46941	3	1014	6	6	0	maternal_health_risk
46955	3	78053	11	9	2	SDSS17
46956	2	2584	15	11	4	seismic-bumps
46960	3	4424	36	19	17	students_dropout_and_academic_success
46963	3	1353	9	0	9	website_phishing
46969	2	7491	86	0	86	NATICUSdroid
46979	2	10885	21	21	0	jm1

datasets with more than one hundred thousand samples, and their dimensionalities range from low-dimensional structured tables to high-dimensional sparse or categorical-heavy datasets.

To ensure the feasibility of downstream evaluations, we filtered the datasets based on their scale, dimensionality, and quality attributes by excluding: (i) Datasets with too few features (features ¡ 5): The structural signals available for representation learning in such datasets are severely limited. (ii) Datasets with an excessively low sample-to-feature ratio: This mitigates the confounding effects of baseline training instability or out-of-memory (OOM) errors on the comparison results under high-dimensional, low-sample settings. (iii) Datasets with a high overall missing rate (missing ¿ 20%): This ensures that the evaluation results are not predominantly driven by the choice of missing value imputation strategies.

B.2 Synthetic Dataset Construction

The synthetic dataset used in Section 5.2 contains 6,000 samples with four numerical features. We first generate a fixed set of contexts $\boldsymbol{u} = (x_1, x_2)$ and completions $\boldsymbol{v} = (x_3, x_4)$. For each $s \in \{-3, -1, 1, 3\}$, we generate 10 contexts by sampling $x_1 \sim \mathcal{N}(0, 1)$ and setting $x_2 = s - x_1$. Completions are generated in the same way using x_3 and x_4. We then combine contexts and completions to form training rows, with a proportion q satisfying $x_1 + x_2 = x_3 + x_4$ and the remainder having different sums. For $q = 0.8$, the oracle CCR scores under the complete reference distribution are approximately 1.16 and -1.32 for supported and suppressed combinations, respectively. We use only a subset of the supported $(\boldsymbol{u}, \boldsymbol{v})$ combinations during pretraining and reserve the remaining combinations for evaluation. For every held-out pair, both $\boldsymbol{u}$ and $\boldsymbol{v}$ appear elsewhere during pretraining, but they are never observed together. We vary the proportion of supported combinations included during pretraining while keeping the number of training samples fixed. The Python implementation of the synthetic dataset construction is illustrated in Fig. 6.

C Details of Implementing Baselines

Common protocol. Because the baselines use different architectures and pretraining objectives, we do not force them to share an identical network, as this would alter their original designs. We retain each baseline's method-specific architecture and follow the configurations recommended by its original paper, using the released implementation whenever available. For each dataset, we repeat pretraining with 100 random seeds. Within each run, all methods share the same data splits and few-shot labeled subsets, with pretraining performed on the unlabeled training split without access to validation or test labels. Under frozen evaluation, each method's pretraining head is discarded, and its representations are evaluated using the same CatBoost classifier. Under fine-tuning, we jointly optimize the pretrained encoder and a two-layer MLP prediction head using AdamW with a learning

```python
# 1. Generate context and completion prototypes
def make_prototypes(n=10, seed=0):
    rng = np.random.default_rng(seed)
    sums = np.array([-3., -1., 1., 3.])
    x1 = rng.normal(size=(4, n))
    x3 = rng.normal(size=(4, n))
    U = np.stack([x1, sums[:, None] - x1], axis=-1)
    V = np.stack([x3, sums[:, None] - x3], axis=-1)
    return U, V

# 2. Split supported pairs into seen and held-out sets
def split_pairs(n=10, coverage=0.6, seed=0):
    rng = np.random.default_rng(seed)
    seen, held_out = [], []
    for g in range(4):
        pairs = [(g, i, j) for i in range(n) for j in range(n)]
        anchors = {(g, i, i) for i in range(n)}
        remaining = list(set(pairs) - anchors)
        rng.shuffle(remaining)

        n_seen = max(n, round(coverage * len(pairs)))
        seen_g = anchors | set(remaining[:n_seen - n])
        seen.extend(seen_g)
        held_out.extend(set(pairs) - seen_g)
    return seen, held_out

# 3. Sample training rows from the seen pairs
def sample_rows(U, V, seen, n_samples=6000, seed=0, q=0.8):
    rng = np.random.default_rng(seed)
    n = U.shape[1]
    n_supported = round(q * n_samples)

    anchors = [(g, i, i) for g in range(4) for i in range(n)]
    X = [np.r_[U[g, i], V[g, j]] for g, i, j in anchors]

    for _ in range(n_supported - len(anchors)):
        g, i, j = seen[rng.integers(len(seen))]
        X.append(np.r_[U[g, i], V[g, j]])
    for _ in range(n_samples - n_supported):
        g_u = rng.integers(4)
        g_v = (g_u + rng.integers(1, 4)) % 4
        i, j = rng.integers(n, size=2)
        X.append(np.r_[U[g_u, i], V[g_v, j]])
    X = np.asarray(X)
    rng.shuffle(X)
    return X
```

Figure 6: Python implementation of the synthetic dataset construction.

rate of 10^{-3} for up to 600 epochs, stopping early if the training loss on the few-shot labeled subset does not improve for 100 consecutive epochs.

CatBoost(Prokhorenkova et al., 2018): is applied directly to the preprocessed input features and does not use unlabeled-data pretraining. It uses the same labeled subsets and CatBoost configuration as the frozen-representation evaluations, providing a strong supervised reference based on the original features. We use CatBoost with 500 iterations with learning rate of 0.1, a maximum tree depth of 3. The same configuration is used for the raw-feature baseline and all frozen-representation evaluations.

VIME(Yoon et al., 2020): In VIME's official implementation [1] , corruptions are randomly selected input features, and it jointly learns to reconstruct their original values and identify the corruption mask. We use the output of its encoder as the downstream representation and discard the reconstruction and mask-prediction heads after pretraining.

SAINT(Somepalli et al., 2021): We follow the released official implementation [2] . SAINT embeds numerical and categorical features and applies attention across both columns and rows. Its self-supervised stage combines contrastive representation alignment with denoising reconstruction. The pretrained transformer output is used as the row representation for downstream evaluation.

SCARF(Bahri et al., 2022): We implement SCARF following its original contrastive version[3] . For each row, SCARF generates a positive view by replacing a random subset of features with values sampled from their empirical marginal distributions. The encoder is pretrained using a contrastive objective between the original and corrupted views, after which its projection head is discarded.

T-JEPA(Thimonier et al., 2025): We follow the official implementation[4] , which partitions the input columns into context and target subsets and trains a predictor to infer the latent representation of the target columns from the context representation. We retain the context encoder after pretraining and discard the target-specific prediction modules.

STUNT(Nam et al., 2023): We use the official implementation[5] . STUNT generates pseudo-tasks from the unlabeled table by selecting columns, constructing pseudo-labels from their values, and meta-learning across the resulting tasks. Its learned encoder is transferred to the downstream few-shot classification tasks using the same labeled subsets as the other methods.

SubTab(Ucar et al., 2021): We follow the official SubTab implementation[6] . SubTab divides the input columns into multiple feature subsets and learns to reconstruct the complete row from each subset. At inference time, representations obtained from the feature subsets are aggregated following the collaborative-inference procedure proposed in the original work.

SwitchTab(Wu et al., 2024): We implement SwitchTab following the pseudocode in the paper. SwitchTab decomposes row representations into mutual and salient components, exchanges the latent components between paired rows, and reconstructs the corresponding inputs. The pretrained encoder representation is used for downstream evaluation.

ICE-T(Tokar and Sanner, 2025): We use the released official implementation code[7]. ICE-T constructs one-column embeddings and contrasts each column with an aggregated representation of its complementary columns. The resulting column embeddings are aggregated into a row-level representation and evaluated under the same downstream protocol as the other SSL baselines.

D Additional Results

D.1 Significance Test

We further examine whether the dataset-level improvements remain consistent across different few-shot labeled subsets. For each dataset and label budget, CCR and the competing method are evaluated over 100 matched runs, and a two-sided paired t-test is applied to their accuracy differences. A comparison is marked as positive when CCR has a higher mean accuracy and negative otherwise. For each baseline and shot, we apply the Holm correction jointly to the dataset-level p-values. The reported #Sig values count the datasets that remain significant at $\alpha = 0.05$ after correction, while the p-value summaries in Table 5 are unadjusted. SAINT comparisons exclude the two datasets on which it runs out of memory.

[1] https://github.com/jsyoon0823/VIME
[2] https://github.com/somepago/saint
[3] https://github.com/clabrugere/pytorch-scarf
[4] https://github.com/jose-melo/t-jepa
[5] https://github.com/jaehyun513/STUNT
[6] https://github.com/AstraZeneca/SubTab
[7] https://github.com/tomastokar/ICET

As shown in Table 5, CCR achieves higher mean accuracy on most datasets across all three label budgets. Significant positive improvements also consistently outnumber significant negative results after the Holm correction. The number of significant gains generally increases from 1-shot to 10-shot, as downstream evaluation becomes more stable with additional labeled examples. Although CCR performs worse on a small number of datasets, only a few of these differences remain significant after correction.

Table 5: Per-dataset paired t-test statistics over 50 datasets. Within every dataset and shot, CCR and the competing method are paired using the same seed. The #Sig counts the results that remain significant after Holm correction. We exclude SAINT OOM results.

Pair	Direction	Count	#Sig	Min p	Q1	Median	Q3	Max p	Mean Δ	Median Δ
1-shot										
CCR / CatBoost	Positive	47	36	4.036×10^{-50}	6.710×10^{-19}	2.100×10^{-6}	0.00312	0.5097	+7.557	+6.228
CCR / CatBoost	Negative	3	1	7.321×10^{-5}	0.06068	0.1213	0.2977	0.4742	-3.094	-3.752
CCR / VIME	Positive	47	30	1.995×10^{-55}	1.136×10^{-15}	8.967×10^{-7}	0.02774	0.6071	+7.018	+5.237
CCR / VIME	Negative	3	1	0.002468	0.01519	0.02791	0.2206	0.4133	-0.650	-0.593
CCR / SAINT	Positive	43	24	7.323×10^{-82}	4.790×10^{-14}	3.471×10^{-5}	0.2613	0.9622	+5.888	+3.712
CCR / SAINT	Negative	5	1	4.808×10^{-4}	0.124	0.1241	0.3403	0.4601	-2.525	-2.812
CCR / SCARF	Positive	40	20	1.123×10^{-66}	1.048×10^{-10}	0.00158	0.1719	0.9191	+4.173	+2.771
CCR / SCARF	Negative	10	2	9.505×10^{-8}	0.0215	0.05241	0.1353	0.725	-2.110	-1.721
CCR / T-JEPA	Positive	47	30	3.675×10^{-79}	2.715×10^{-12}	6.016×10^{-5}	0.02035	0.743	+6.694	+4.165
CCR / T-JEPA	Negative	3	2	2.063×10^{-6}	9.927×10^{-4}	0.001983	0.08656	0.1711	-1.821	-1.498
CCR / STUNT	Positive	41	26	3.843×10^{-27}	4.103×10^{-8}	3.068×10^{-4}	0.02689	0.8389	+5.303	+3.627
CCR / STUNT	Negative	9	3	8.798×10^{-13}	1.160×10^{-5}	0.01469	0.3078	0.7823	-2.503	-1.765
CCR / SubTab	Positive	41	33	7.075×10^{-56}	1.363×10^{-26}	3.314×10^{-9}	5.389×10^{-4}	0.8859	+8.029	+5.984
CCR / SubTab	Negative	9	2	7.600×10^{-6}	0.0933	0.2659	0.5272	0.9711	-2.059	-0.922
CCR / SwitchTab	Positive	39	29	2.622×10^{-49}	3.642×10^{-16}	5.467×10^{-10}	0.004313	0.3948	+8.003	+5.761
CCR / SwitchTab	Negative	11	6	3.341×10^{-11}	1.239×10^{-4}	0.002936	0.04717	0.81	-2.862	-2.646
CCR / ICE-T	Positive	40	27	8.842×10^{-33}	3.029×10^{-18}	1.985×10^{-5}	0.02964	0.8697	+6.195	+3.711
CCR / ICE-T	Negative	10	5	9.500×10^{-11}	3.353×10^{-6}	0.005716	0.4189	0.987	-3.484	-2.037
5-shot										
CCR / CatBoost	Positive	43	31	6.483×10^{-50}	8.650×10^{-20}	1.872×10^{-8}	0.01462	0.7992	+5.223	+3.806
CCR / CatBoost	Negative	7	3	1.832×10^{-7}	6.289×10^{-6}	0.02653	0.04058	0.516	-2.795	-2.032
CCR / VIME	Positive	47	44	2.324×10^{-84}	2.094×10^{-36}	2.057×10^{-14}	1.070×10^{-8}	0.202	+8.882	+5.492
CCR / VIME	Negative	3	1	0.008314	0.3393	0.6702	0.7467	0.8232	-0.971	-0.167
CCR / SAINT	Positive	44	27	6.096×10^{-90}	8.016×10^{-18}	2.916×10^{-5}	0.148	0.9638	+5.414	+2.947
CCR / SAINT	Negative	4	1	1.253×10^{-4}	0.008353	0.02998	0.1243	0.3504	-1.896	-2.191
CCR / SCARF	Positive	47	34	3.541×10^{-81}	1.981×10^{-28}	3.592×10^{-11}	0.005869	0.5315	+6.054	+3.941
CCR / SCARF	Negative	3	1	0.003113	0.01557	0.02802	0.1053	0.1826	-1.452	-1.305
CCR / T-JEPA	Positive	42	32	1.394×10^{-59}	2.143×10^{-32}	3.969×10^{-9}	0.002714	0.9762	+6.921	+4.129
CCR / T-JEPA	Negative	8	2	5.277×10^{-15}	0.008904	0.04341	0.2269	0.7956	-1.825	-1.595
CCR / STUNT	Positive	44	31	8.634×10^{-52}	7.832×10^{-16}	4.895×10^{-6}	0.02004	0.8857	+4.771	+3.481
CCR / STUNT	Negative	6	3	1.130×10^{-4}	0.001155	0.01572	0.3993	0.8296	-1.247	-0.758
CCR / SubTab	Positive	45	38	9.473×10^{-87}	2.268×10^{-30}	3.492×10^{-14}	3.395×10^{-5}	0.6835	+7.538	+4.616
CCR / SubTab	Negative	5	1	5.741×10^{-4}	0.3938	0.8189	0.8402	0.9418	-0.873	-0.204
CCR / SwitchTab	Positive	40	37	4.212×10^{-81}	3.259×10^{-35}	1.838×10^{-20}	8.678×10^{-7}	0.6784	+10.348	+7.719
CCR / SwitchTab	Negative	10	6	1.091×10^{-17}	2.401×10^{-7}	2.350×10^{-4}	0.1439	0.7368	-1.632	-1.430
CCR / ICE-T	Positive	43	34	5.154×10^{-56}	3.681×10^{-24}	3.294×10^{-10}	0.00169	0.814	+6.698	+3.806
CCR / ICE-T	Negative	7	2	2.295×10^{-8}	0.002199	0.005199	0.02412	0.4228	-2.553	-1.526
10-shot										
CCR / CatBoost	Positive	44	33	1.298×10^{-69}	1.223×10^{-28}	4.156×10^{-14}	0.001161	0.7444	+4.643	+3.626
CCR / CatBoost	Negative	6	3	2.294×10^{-31}	8.273×10^{-4}	0.009123	0.01688	0.4213	-2.319	-1.873
CCR / VIME	Positive	47	47	1.783×10^{-94}	2.736×10^{-45}	2.905×10^{-30}	2.189×10^{-12}	0.008301	+8.959	+5.679
CCR / VIME	Negative	3	1	0.01658	0.1128	0.209	0.4577	0.7065	-0.178	-0.187
CCR / SAINT	Positive	46	30	7.854×10^{-91}	6.335×10^{-21}	8.084×10^{-8}	0.03729	0.8707	+4.656	+2.670
CCR / SAINT	Negative	2	2	9.719×10^{-83}	7.279×10^{-4}	0.001456	0.002184	0.002912	-1.285	-1.285
CCR / SCARF	Positive	48	39	5.592×10^{-117}	7.671×10^{-36}	4.288×10^{-17}	1.040×10^{-4}	0.9157	+6.611	+3.989
CCR / SCARF	Negative	2	2	1.026×10^{-4}	0.001286	0.00247	0.003653	0.004836	-1.332	-1.332
CCR / T-JEPA	Positive	42	31	1.852×10^{-45}	5.625×10^{-26}	8.339×10^{-11}	0.006104	0.9037	+5.399	+3.575
CCR / T-JEPA	Negative	8	4	5.878×10^{-31}	4.046×10^{-9}	0.006885	0.1763	0.9009	-1.558	-1.264
CCR / STUNT	Positive	47	39	7.937×10^{-80}	3.208×10^{-28}	2.636×10^{-14}	3.673×10^{-5}	0.6005	+5.579	+4.244
CCR / STUNT	Negative	3	1	0.003316	0.0707	0.1381	0.4421	0.746	-1.002	-0.248
CCR / SubTab	Positive	47	43	8.694×10^{-117}	5.725×10^{-45}	2.091×10^{-20}	3.569×10^{-9}	0.4974	+7.732	+6.102
CCR / SubTab	Negative	3	1	0.006989	0.008843	0.0107	0.3303	0.65	-0.196	-0.213
CCR / SwitchTab	Positive	43	42	1.306×10^{-94}	1.119×10^{-43}	2.127×10^{-29}	1.511×10^{-15}	0.03072	+10.457	+7.962
CCR / SwitchTab	Negative	7	6	6.011×10^{-82}	1.391×10^{-49}	2.693×10^{-45}	0.008178	0.06866	-1.511	-1.979
CCR / ICE-T	Positive	44	40	6.179×10^{-91}	2.679×10^{-34}	3.167×10^{-14}	8.859×10^{-9}	0.09552	+7.162	+4.750
CCR / ICE-T	Negative	6	4	3.308×10^{-23}	1.435×10^{-7}	0.003572	0.2058	0.4715	-3.029	-1.517

D.2 Hyperparameter Sensitivity and Evaluator Robustness

We examine the sensitivity of CCR to the embedding dimension and the number of marginal candidates K, as well as its robustness across different downstream classifiers. For each setting, we report the average accuracy change relative to the default configuration across 50 datasets. The default values are an embedding dimension of 128, $K = 4$, and a mask ratio of 0.3. For the downstream evaluation, we report the accuracy gain of CCR representations over the preprocessed raw features using LR, MLP, and CatBoost.

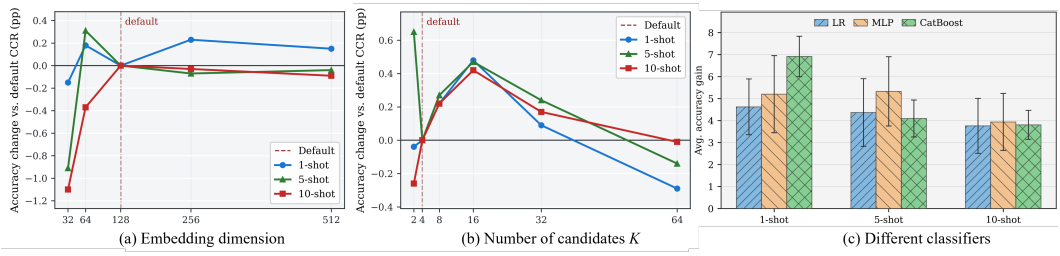


Figure 7: Hyperparameter sensitivity and evaluator robustness of CCR. (a) and (b) show accuracy changes from the default configuration; dashed lines mark the default values. (c) shows accuracy gains over raw features; error bars denote standard error across 50 datasets.

Table 6: Effect of CCR pretraining relative to architecture-matched controls over 50 datasets. Entries report the mean paired accuracy gain in percentage points; parentheses give the number of datasets on which pretrained CCR performs better.

Protocol	Control	1-shot	5-shot	10-shot
Frozen	Random Encoder	+2.76 (44/50)	+2.04 (40/50)	+1.64 (37/50)
Fine-tuning	Scratch	+1.97 (40/50)	+1.48 (37/50)	+1.01 (34/50)

As shown in Fig. 7, CCR is generally stable over a broad range of hyperparameter values. For the embedding dimension, dimensions between 64 and 512 produce similar performance, while reducing the dimension to 32 causes a clear drop in the 5- and 10-shot settings. This suggests that a very small representation may limit the information retained by the encoder, whereas increasing the dimension beyond 128 provides no consistent benefit. The performance is also relatively stable with respect to K. Using more than four marginal candidates can slightly improve some settings, but the gains are small and do not remain consistent across label budgets. A very large value, such as $K = 64$, reduces performance, especially in the 1-shot setting. We therefore use $K = 4$, which provides a good balance between performance and training cost.

Robustness to Downstream Evaluators. To examine whether the gains of CCR depend on a particular downstream evaluator, we further consider logistic regression (LR), a two-layer MLP, and CatBoost. These models represent linear, neural, and tree-based predictive biases, respectively. For each evaluator, we compare models trained on the preprocessed raw features against models trained with CCR. As shown in Fig. **??**, CCR achieves positive average gains with LR, MLP, and CatBoost in the limited label scenario. The gains are generally larger in lower-shot settings. This shows that the improvement of CCR is not tied to a particular downstream evaluator.

D.3 Does pretraining help beyond the encoder architecture?

To verify that the improvements arise from self-supervised pretraining rather than the encoder architecture itself, we introduce two architecture-matched controls. Under frozen evaluation, *Random Encoder* applies the downstream evaluator to representations produced by an untrained encoder with the same architecture and output dimension as CCR. Under fine-tuning, *Scratch* randomly initializes the same encoder and prediction head and trains the entire model using only the corresponding k-shot labeled subset. The labeled subsets, initialization seeds, downstream architecture, and optimization budget are matched to those used by pretrained CCR.

As shown in Table 6, pretrained CCR consistently outperforms both controls. The advantage is largest in the 1-shot regime and remains positive as the label budget increases. In particular, the improvement over the frozen random encoder shows that the gain cannot be explained by a random nonlinear projection, while the improvement over end-to-end training from scratch indicates that the encoder initialization learned from unlabeled data remains beneficial even when the full model is subsequently updated.

D.4 Verification of Density-Ratio Recovery

In this part, we use a binary distribution to examine whether CCR can recover the density ratio described in Proposition 1. We sample $U \sim \text{Bernoulli}(0.5), A \sim \text{Bernoulli}(0.5), E \sim \text{Bernoulli}(\epsilon)$, and define $B = A \oplus U \oplus E$. We also add eight independent binary noise features to the context. These features are independent of U, A, and B. By construction, observing U does not change the probability of either target variable on its own:

$$p(A \mid U) = p(A) = 0.5, \qquad p(B \mid U) = p(B) = 0.5. \tag{33}$$

Instead, U changes how A and B occur together. When $U = 0$, A and B are more likely to have the same value; when $U = 1$, they are more likely to have different values. All four combinations of (A, B) are equally frequent before U is observed, giving $p(A, B) = 1/4$. The true log density ratio is therefore:

$$r(U, A, B) = \log \frac{p(A, B \mid U)}{p(A, B)} = \begin{cases} \log\big(2(1-\epsilon)\big), & B = A \oplus U, \\ \log(2\epsilon), & B \neq A \oplus U. \end{cases} \tag{34}$$

For every $\epsilon < 0.5$, combinations that follow the relation $B = A \oplus U$ have a positive score, while the remaining combinations have a negative score. We use U and the independent noise features as the context and treat (A, B) as a joint target. To construct a negative sample, we keep the context fixed and replace (A, B) with values taken together from another randomly selected row. Sampling both target values from the same row preserves their joint distribution while breaking their dependence on the current context. When using K negative samples for each positive sample, we average the positive and negative losses separately so that the two groups contribute equally. This prevents the learned score from acquiring an additional $-\log K$ shift. We use 20,000 training samples, 5,000 validation samples, and five random seeds for each value of ϵ.

For metrics, here we consider: (i) Ratio RMSE is the root mean squared difference between the learned score $S(u, (a, b); M)$ and the corresponding theoretical value $r(U, A, B)$ defined above, over the eight possible (U, A, B) configurations. (ii) Slope is obtained by regressing the learned scores on the theoretical scores with an intercept; a slope of one indicates correct score scaling. (iii) Jensen–Shannon divergence (JS) between the true conditional distribution $p(A, B \mid U)$ and the distribution reconstructed from the learned scores, where $\widehat{p}(a, b \mid u) \propto p(a, b) \exp(S(u, (a, b); M))$. The reconstructed distribution is normalized over the four possible (a, b) combinations, and JS is averaged over $U = 0$ and $U = 1$.

Table 7: Mean density-ratio recovery results over five random seeds under different noise levels ϵ.

ϵ	True score (match)	True score (mismatch)	Ratio RMSE	Slope	JS
0.05	0.642	−2.303	0.101	1.004	0.0003
0.10	0.588	−1.609	0.087	1.006	0.0005
0.20	0.470	−0.916	0.048	0.990	0.0001
0.30	0.336	−0.511	0.043	0.992	0.0001
0.40	0.182	−0.223	0.034	0.993	0.0001

Table 7 shows that the learned scores closely match the theoretical values across all tested noise levels. The mean slope remains close to one, and the Jensen–Shannon divergence between the reconstructed and true conditional distributions is at most 5×10^{-4}, showing that CCR recovers both the scale of the scores and the resulting conditional distribution. Finally, the RMSE decreases at larger values of ϵ partly because the true scores move closer to zero.

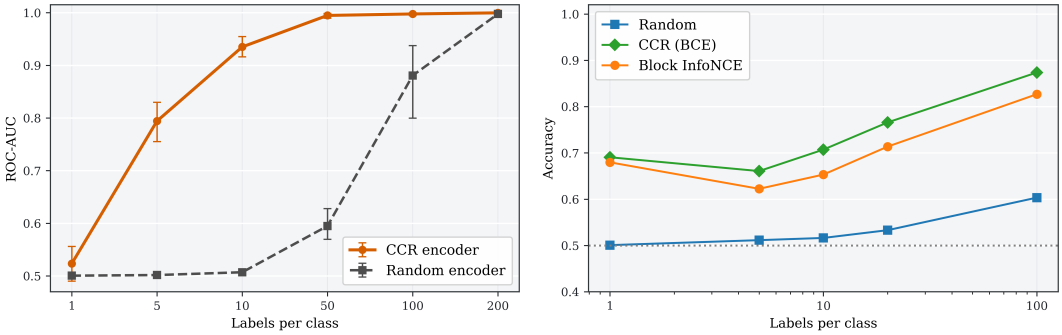
D.5 Does the encoder preserve the learned feature dependencies?

In Sec. 4.2, we characterize the score produced jointly by the encoder and scorer, whereas downstream evaluation discards the scorer and retains only the encoder. Here, we examine whether the feature dependency learned during pretraining remains accessible from the encoder representation.

We use the same synthetic distribution as in Appendix D.4 and construct a binary task that asks whether a pair (A, B) follows the relation specified by the context variable U. After CCR pretraining, we remove the scorer and encode each complete row using the frozen encoder. We then train

a 3-layer MLP for classification, with different numbers of labeled examples per class. As a control, we apply the same procedure to an identically parameterized encoder without pretraining. Both encoders are frozen, and the two probes use the same labeled subsets and optimization settings.

Fig. 8a shows that the dependency can be recovered from the frozen CCR encoder with substantially fewer labels than from the random encoder. The difference is small in the 1-shot setting, becomes clear from 5 to 100 shots, and disappears when 200 labeled examples per class are available. This convergence confirms that the relation is learnable from both representations given sufficient supervision; CCR pretraining primarily makes it more accessible in the low-label regime. The result therefore shows that useful information learned during pretraining remains in the retained encoder after the scorer is removed.



(a) Relation recovery from full-row representations

(b) Context recovery from context representations

Figure 8: Few-shot recovery of relation-relevant information from frozen encoders after discarding the pretraining scorer. (a) A three-layer MLP predicts whether a completion follows the context-dependent relation. (b) A logistic regression probe predicts $U = C_1 \oplus C_2$ from the context representation. Error bars show 95% hierarchical bootstrap confidence intervals.

Recovering the context variable from frozen representations. We further examine whether learning context–completion compatibility makes the variable governing this relation more accessible from the context representation. We retain the noisy-XOR construction in Appendix D.4, $B = A \oplus U \oplus E$, but replace the explicit context variable U with two independent binary features C_1 and C_2 satisfying $U = C_1 \oplus C_2$. This preserves the joint distribution of (U, A, B) while requiring the encoder to extract U from a relation between context features. The eight independent binary noise features are retained. After pretraining, we discard the scorer, freeze the context encoder, and train a logistic regression probe to predict U using varying numbers of labels per class. We compare CCR with an identically initialized random encoder and a block InfoNCE variant that shares CCR's encoder, scorer, positive–negative pairs, and training budget. All methods use the same labeled subsets for evaluation.

Fig. 8b shows that CCR consistently improves the accessibility of U across the evaluated label budgets. Its mean accuracy with one label per class already exceeds that of the random encoder with 100 labels per class, indicating substantially improved label efficiency. Since the probe observes only the frozen context representation, this gain shows that the compatibility-learning signal shapes the retained encoder, making the variable governing the completion relation easier to recover with a linear classifier. Block InfoNCE exhibits a similar improvement, while CCR attains higher accuracy at most label budgets. Together, these results support context–completion pretraining as an effective way to expose relation-relevant information in the context representation.

D.6 Comparing Pretraining Objectives with a Fixed Backbone

To isolate the effect of the pretraining objective, we use the same encoder architecture, representation dimension, masking distribution, optimizer, batch size, and training budget for all methods. We compare five objectives: (i) masked reconstruction, which applies MSE to numerical features and cross-entropy (CE) to categorical features; (ii) instance discrimination using InfoNCE on the same corrupted views; (iii) binned feature prediction, which discretizes numerical features into quantile

bins and applies CE to both the discretized numerical features and the original categorical features; (iv) CCR, which performs conditional–marginal discrimination using binary cross-entropy (BCE); and (v) CCR-Cal, an optional variant that uses reliability-weighted BCE. All objective-specific prediction or scoring heads are discarded before downstream evaluation.

Table 8: Comparison of pretraining objectives using the same backbone across 50 datasets. Values are mean accuracy improvements relative to no pretraining.

	Frozen encoder			Fine-tuned encoder		
Loss function	1-shot	5-shot	10-shot	1-shot	5-shot	10-shot
No pretraining	0.00	0.00	0.00	0.00	0.00	0.00
Masked reconstruction (MSE + CE)	0.86	0.63	0.45	0.49	0.36	0.25
Instance discrimination (InfoNCE)	1.23	0.94	0.68	0.81	0.60	0.43
Binned feature prediction (CE)	1.51	1.67	0.85	1.02	0.73	0.54
CCR (BCE)	2.76	2.04	**1.64**	1.97	1.48	**1.01**
CCR-Cal (weighted BCE)	**2.96**	**2.37**	1.52	**2.42**	**1.49**	0.98

As shown in Table 8, CCR provides larger gains than reconstruction, instance discrimination, and binned feature prediction under the same backbone. CCR-Cal produces modest additional improvements in four of the six settings, while slightly decreasing 10-shot fine-tuning accuracy. These results indicate that the main gain comes from conditional–marginal discrimination, whereas calibration has a smaller, setting-dependent effect.

Comparison with block InfoNCE. We further explore an alternative objective for learning the conditional-to-marginal density ratio to examine how the choice of objective affects representation quality. Specifically, we formulate a block InfoNCE objective that identifies the observed completion among $K + 1$ candidates for each context. We keep the encoder, scorer, feature partitions, and sampled completions identical to those of CCR, and evaluate both objectives on all 50 datasets using frozen encoders and CatBoost over 10 different seeds.

Table 9: Accuracy(%) comparison with block InfoNCE across 50 datasets over 10 different random seeds. ΔAcc denotes CCR minus block InfoNCE.

Shot	CCR	Block InfoNCE(%)	ΔAcc (pp)
1	57.54	57.01	+0.53
5	68.42	67.97	+0.44
10	72.25	72.16	+0.09

Table 9 shows that CCR achieves slightly higher average accuracy across all three label budgets. The gains are larger in the 1-shot and 5-shot settings, while the two objectives perform similarly at 10-shot. The results of both objectives suggest that context–completion discrimination may provide a promising learning objective for tabular representation learning.

D.7 Analysis of CCR-Cal with Limited Unlabeled Data

Here, we conduct an experiment on eight datasets selected before examining their calibration results. As summarized in Table 10, the selected datasets cover numerical-only, categorical-only, and mixed inputs, as well as a wide range of sample-to-dimension ratios. HAR and DNA represent datasets with few training rows relative to their feature dimensions, whereas letter and connect-4 represent the opposite case. The remaining four datasets lie between these two extremes and move toward the limited-data region when fewer unlabeled rows are used.

For each dataset, we randomly order the unlabeled training rows once and use the first 10%, 25%, 50%, or 100% for pretraining. Thus, the 10% subset is included in the 25% subset, which is further included in the 50% and 100% subsets. Across the four settings, we keep the input features, labeled few-shot samples, validation set, and test set unchanged. Therefore, the only change is the amount of unlabeled data used to train CCR.

Table 10: Datasets selected for the calibration experiment. Ratio denotes the number of unlabeled training rows per input feature.

ID	Dataset	Feature type	Rows per feature	
			10% data	100% data
1478	HAR	Numerical	1.3	12.9
40670	DNA	Categorical	1.2	12.4
40499	Texture	Numerical	9.6	96.3
46969	NATICUSdroid	Categorical	6.1	61.0
46960	Students dropout	Mixed	8.6	86.0
46937	In-vehicle coupon	Mixed	37.0	369.9
6	Letter	Numerical	87.5	875.0
40668	Connect-4	Categorical	112.6	1125.9

Table 11: Accuracy gain of CCR-Cal over CCR with different amounts of unlabeled training data. For each dataset, results are first averaged over the 1-, 5-, and 10-shot settings. Bracketed ranges denote 95% confidence intervals across the eight datasets.

Unlabeled data used	Accuracy gain (pp)	Wins/Losses
10%	$+1.08$ $[0.61, 1.54]$	7/1
25%	$+0.71$ $[0.36, 1.07]$	7/1
50%	$+0.38$ $[0.13, 0.64]$	6/2
100%	$+0.14$ $[-0.06, 0.34]$	4/4

As shown in Table 11, calibration helps more when less unlabeled data are available. With only 10% of the unlabeled rows, it improves accuracy by 1.08 percentage points and performs better on seven of the eight datasets. As more unlabeled rows are added, the gain gradually decreases. With the full unlabeled training set, the average gain is only 0.14 percentage points, and calibration wins on four datasets and loses on four.

Furthermore, we divide the 50 evaluated datasets according to N_{train}/d, where N_{train} is the number of unlabeled training instances and d is the number of input features before one-hot encoding. We divide the 50 datasets into low, medium, and high groups and report the accuracy gain of calibrated CCR over its uncalibrated variant. Table 12 shows that the benefit of reliability calibration decreases

Table 12: Effect of reliability calibration across different sample-to-dimension ratios. ΔAcc denotes CCR-Cal minus CCR.

Group	N_{train}/d	#Datasets	1-shot ΔAcc	5-shot ΔAcc	10-shot ΔAcc
Low	$N_{\text{train}}/d < 50$	17	0.36	0.50	0.14
Medium	$50 \leq N_{\text{train}}/d \leq 500$	24	0.20	0.36	-0.16
High	$N_{\text{train}}/d > 500$	9	-0.05	-0.12	-0.53

as the number of training instances relative to the feature dimension increases. Calibration provides the largest improvement in the low-ratio group, where the available samples provide limited coverage of the feature space. Its benefit becomes smaller in the medium- and high-ratio groups. This trend is consistent across the three few-shot settings.

D.8 Comparison with Tabular Foundation Models

We further compare CCR with two tabular foundation models, TabPFN Hollmann et al. (2025) and TabICL Qu et al. (2025). We use the newest v3 checkpoint for TabPFN and the v2 checkpoint for TabICL, which were the latest weights available from their respective official repositories at the time of our experiments. Both models are evaluated using their official inference configurations. All methods use the same data splits and the same labeled subsets under 100 matched random seeds. CCR uses the unlabeled training dataset for pretraining, whereas TabPFN and TabICL rely on their externally pretrained checkpoints.

Table 13 shows that, in the 1-shot regime, CCR achieves the best accuracy on 43 of the 50 datasets. As the label budget increases, the gap narrows substantially, while CCR retains the best average rank in both the 5-shot and 10-shot settings. Overall, CCR demonstrates strong and consistent performance across few-shot regimes.

Table 13: Comparison with tabular foundation models across 50 datasets (↓ for rank).

	1-shot		5-shot		10-shot	
Method	Rank	#Best	Rank	#Best	Rank	#Best
TabPFN_v3	2.52	4	2.14	13	2.10	12
TabICL_v2	2.26	3	2.04	11	2.06	12
CCR	**1.22**	**43**	**1.82**	**26**	**1.84**	**26**

D.9 Computational Complexity

In this part, we analyze the computational complexity of CCR and its scalability. Consider a batch of B rows, with one sampled mask and K negative completions per row. Let C_E denote the cost of encoding one masked input and C_S the cost of projecting and scoring one context–completion pair. For each row, CCR encodes the context, the observed completion, and K sampled completions, and then evaluates $K + 1$ pairs. Without reusing candidate encodings, the resulting per-batch forward cost is $O(B(K + 2)C_E + B(K + 1)C_S)$, and backpropagation has the same asymptotic order. For a fixed input dimensionality and network architecture, this cost grows linearly with both the batch size and the number of sampled candidates.

Although a larger table increases the encoding cost, CCR never enumerates the possible combinations of missing feature values or computes a conditional normalization constant. Its computation therefore depends on the sampled candidates rather than the size of the completion space. At downstream evaluation, we discard the projection layers and scorer and retain only the encoder, giving an encoding cost of $O(C_E)$ per instance. These properties make CCR practical for learning from multi-column completions while keeping its pretraining cost controllable through K.

D.10 Limitations and Future Work

High-Dimensional Estimation. Although CCR estimates the density ratio directly, recovering complex dependencies from finite samples can become challenging as dimensionality increases. Such statistical difficulties also arise in other density-ratio and contrastive estimators (Rhodes et al., 2020; Choi et al., 2022). Recent Transformer-based density and score estimators, such as DiSco-Former (Ilin et al., 2026), suggest a promising direction for mitigating this limitation. Adapting their sample-based attention and cross-distribution pretraining to CCR could improve statistical efficiency on high-dimensional tables.

Adaptive Mask Sampling. CCR samples random column masks without adapting to the informativeness of different context–completion partitions. Correlation-aware gate sampling in SEFS (Lee et al., 2022) offers a related example of incorporating feature structure into self-supervision. Although our ablations favor random masking over Pearson-guided selection, adaptive policies could further improve training efficiency by prioritizing informative partitions while maintaining broad coverage of column subsets.

Cross-Table Pretraining. The current implementation uses a fixed feature schema and is pretrained separately on each table. Extending CCR to shared pretraining requires representations that accommodate different column identities and feature types. Column-description-based encoding in TransTab (Wang and Sun, 2022) and shared-backbone pretraining in XTab (Zhu et al., 2023) offer possible routes for applying CCR across heterogeneous tables.

D.11 AVERAGE RANKINGS

For each dataset, methods are ranked by accuracy, with OOM assigned the last rank, and then averaged across all 50 datasets. CCR ranks first in every few-shot setting under both evaluation protocols, consistent with the representative results in Table 1.

Table 14: Complete average accuracy ranks(↓) across 50 datasets, lower is better.

	1-shot		5-shot		10-shot	
Method	Frz	FT	Frz	FT	Frz	FT
CatBoost	7.1	–	5.3	–	4.8	–
VIME	6.6	7.7	7.4	7.6	7.4	7.5
SAINT	5.4	5.2	5.1	4.8	5.0	4.4
SCARF	4.4	4.5	5.9	5.0	5.8	4.8
T-JEPA	6.2	3.4	5.6	3.1	4.6	2.8
STUNT	5.1	4.9	5.0	4.5	5.8	5.4
SubTab	6.5	6.8	6.4	6.7	6.8	6.2
SwitchTab	6.1	5.8	6.5	6.2	7.0	6.7
ICE-T	5.4	4.7	5.6	5.2	6.0	5.1
CCR	**2.2**	**1.9**	**2.1**	**1.9**	**1.8**	**2.1**

D.12 DETAILED RESULTS

Table 15: 1-shot results. Each entry is mean ± standard deviation over repeated runs; the best and second-best methods for each dataset are shown in bold and underlined, respectively.

OpenML ID	CatBoost	VIME	SAINT	SCARF	T-JEPA	STUNT	SubTab	SwitchTab	ICE-T	CCR
6	25.72 ± 3.27	19.18 ± 2.35	23.91 ± 1.56	18.83 ± 2.32	25.37 ± 2.56	25.19 ± 2.91	21.21 ± 1.99	23.44 ± 3.48	22.74 ± 1.96	**26.19 ± 2.98**
12	61.88 ± 4.19	55.71 ± 6.72	63.25 ± 6.05	62.45 ± 8.55	62.60 ± 7.18	**71.00 ± 7.69**	65.38 ± 6.65	67.10 ± 6.74	62.05 ± 6.00	64.45 ± 7.41
15	84.43 ± 11.16	91.56 ± 5.76	93.21 ± 1.02	92.14 ± 5.70	89.71 ± 6.64	93.79 ± 1.47	92.57 ± 4.52	93.50 ± 3.82	**94.79 ± 1.17**	92.79 ± 4.55
16	40.62 ± 7.89	40.28 ± 5.02	39.80 ± 8.10	56.80 ± 5.04	48.33 ± 4.96	**63.65 ± 7.07**	50.82 ± 8.11	63.35 ± 6.46	62.70 ± 6.81	60.73 ± 7.69
22	34.90 ± 3.51	40.09 ± 5.73	38.80 ± 3.39	36.48 ± 4.09	44.15 ± 4.60	44.00 ± 4.62	41.52 ± 4.82	**45.23 ± 4.87**	33.48 ± 2.69	41.38 ± 4.23
28	56.57 ± 6.47	61.15 ± 5.24	54.36 ± 4.95	70.46 ± 3.29	59.56 ± 4.22	65.88 ± 5.16	56.05 ± 4.25	69.89 ± 5.91	62.14 ± 5.51	**71.23 ± 3.98**
29	59.57 ± 9.93	60.04 ± 8.34	62.25 ± 8.71	61.67 ± 15.83	55.80 ± 10.49	65.07 ± 9.43	55.07 ± 6.61	49.86 ± 5.54	55.43 ± 8.65	**66.09 ± 8.02**
32	52.21 ± 3.71	58.43 ± 6.34	57.27 ± 3.99	59.41 ± 3.02	58.12 ± 4.30	**63.02 ± 4.44**	53.04 ± 4.78	61.36 ± 5.52	55.23 ± 5.30	62.21 ± 3.39
37	58.64 ± 9.49	59.56 ± 11.92	61.76 ± 4.41	**62.86 ± 9.27**	60.19 ± 9.74	58.25 ± 6.82	62.47 ± 7.20	52.14 ± 13.48	59.81 ± 10.75	61.88 ± 8.14
38	46.94 ± 25.49	55.10 ± 19.57	73.13 ± 11.07	73.27 ± 5.86	59.60 ± 17.83	58.89 ± 15.75	**75.03 ± 12.32**	38.97 ± 14.05	48.19 ± 26.75	68.38 ± 7.07
44	72.58 ± 4.76	59.65 ± 10.74	73.07 ± 4.21	78.72 ± 9.38	69.02 ± 4.99	69.51 ± 8.64	72.31 ± 7.45	78.30 ± 7.24	81.30 ± 6.15	**81.45 ± 6.90**
46	40.58 ± 4.40	44.08 ± 7.61	44.15 ± 7.52	63.61 ± 8.69	41.32 ± 5.44	55.72 ± 19.02	51.49 ± 6.62	40.45 ± 4.50	52.93 ± 8.62	**71.29 ± 11.67**
50	49.38 ± 6.37	54.23 ± 7.97	52.86 ± 6.38	54.64 ± 8.35	51.41 ± 7.35	49.32 ± 7.42	51.25 ± 6.41	49.95 ± 5.06	**61.46 ± 11.93**	53.07 ± 8.26
182	61.61 ± 7.74	62.33 ± 10.33	59.13 ± 7.03	64.07 ± 10.96	63.23 ± 8.40	64.42 ± 8.79	61.58 ± 8.53	66.74 ± 9.77	57.62 ± 6.69	**67.57 ± 9.35**
458	72.31 ± 9.33	74.93 ± 8.20	80.18 ± 7.16	**92.96 ± 2.42**	79.94 ± 9.65	86.39 ± 8.07	83.85 ± 8.42	91.95 ± 4.38	91.12 ± 5.15	89.05 ± 6.75
1067	63.10 ± 23.30	66.09 ± 23.17	64.10 ± 22.28	63.29 ± 21.61	66.21 ± 22.84	61.52 ± 21.84	61.94 ± 21.63	62.01 ± 23.42	61.68 ± 23.34	**73.98 ± 9.29**
1068	65.68 ± 20.58	62.68 ± 22.61	65.72 ± 19.97	63.56 ± 23.71	61.53 ± 22.85	63.45 ± 20.76	63.60 ± 19.93	**70.09 ± 20.38**	63.96 ± 21.72	63.92 ± 17.51
1468	46.02 ± 5.14	18.11 ± 4.70	OOM	37.13 ± 7.45	26.34 ± 5.94	33.24 ± 15.46	19.77 ± 3.50	34.40 ± 7.51	**48.84 ± 5.95**	42.27 ± 6.52
1478	46.55 ± 6.76	**57.32 ± 7.37**	OOM	42.24 ± 6.70	53.67 ± 7.49	51.45 ± 7.31	55.38 ± 5.59	45.95 ± 8.09	44.36 ± 11.12	57.13 ± 6.67
1480	53.59 ± 13.04	54.17 ± 11.77	50.00 ± 11.97	**58.70 ± 5.88**	53.93 ± 13.60	54.02 ± 8.32	49.74 ± 14.46	53.93 ± 13.19	55.38 ± 11.78	56.84 ± 10.41
1485	48.25 ± 2.60	50.10 ± 2.85	51.22 ± 2.03	51.97 ± 3.17	52.69 ± 1.35	50.98 ± 2.79	50.92 ± 2.71	50.67 ± 2.87	52.77 ± 7.14	**53.36 ± 3.49**
1489	62.54 ± 9.37	61.37 ± 8.43	62.67 ± 6.26	62.22 ± 3.98	60.14 ± 10.13	59.49 ± 12.55	59.77 ± 11.58	60.09 ± 14.25	58.15 ± 16.31	**64.26 ± 10.10**
1494	60.56 ± 8.88	56.66 ± 13.25	62.34 ± 8.26	62.09 ± 7.28	58.39 ± 9.05	61.43 ± 9.03	58.72 ± 9.58	58.72 ± 8.63	**67.16 ± 4.30**	62.56 ± 7.35
1501	38.25 ± 4.03	44.26 ± 5.99	43.23 ± 4.96	42.98 ± 6.91	26.96 ± 2.57	46.58 ± 3.88	41.66 ± 4.59	41.88 ± 4.16	47.24 ± 3.81	**48.62 ± 3.78**
4534	59.10 ± 8.04	61.32 ± 9.09	58.00 ± 8.59	64.22 ± 12.54	59.67 ± 8.39	61.22 ± 10.92	60.60 ± 11.97	64.57 ± 12.67	62.40 ± 14.01	**66.97 ± 7.19**
4538	25.12 ± 6.07	24.91 ± 6.26	25.25 ± 6.04	**26.81 ± 6.09**	**26.81 ± 7.22**	25.83 ± 7.05	26.21 ± 5.98	25.79 ± 7.41	26.11 ± 7.04	25.61 ± 5.68
6332	56.39 ± 11.48	56.02 ± 9.92	53.09 ± 11.14	51.85 ± 8.01	55.37 ± 9.31	53.61 ± 11.48	51.57 ± 10.40	50.65 ± 4.12	58.06 ± 12.02	**59.44 ± 6.58**
23381	50.00 ± 6.39	51.58 ± 6.16	51.14 ± 4.25	51.55 ± 6.10	52.80 ± 6.11	45.90 ± 6.28	47.20 ± 6.18	51.80 ± 7.71	52.30 ± 7.94	**55.98 ± 5.46**
40499	53.55 ± 5.75	60.87 ± 6.02	60.09 ± 4.65	49.44 ± 2.67	61.75 ± 6.13	63.49 ± 3.42	55.68 ± 2.91	**66.33 ± 4.20**	57.33 ± 3.89	64.81 ± 3.42
40668	33.08 ± 3.53	36.05 ± 7.61	34.90 ± 7.55	31.30 ± 5.85	37.33 ± 5.41	36.93 ± 5.88	34.44 ± 7.88	34.46 ± 4.85	34.92 ± 7.02	**38.82 ± 5.19**
40670	41.94 ± 6.48	48.32 ± 9.37	48.28 ± 10.26	60.59 ± 8.42	49.28 ± 4.04	49.51 ± 16.58	41.55 ± 6.35	44.12 ± 4.84	53.15 ± 7.40	**61.85 ± 7.95**
40979	54.38 ± 7.03	60.04 ± 6.92	24.79 ± 2.84	60.45 ± 7.96	57.35 ± 6.28	**67.92 ± 8.40**	57.20 ± 5.84	56.80 ± 6.58	60.70 ± 6.34	61.77 ± 5.13
40983	48.37 ± 15.83	49.24 ± 21.72	52.37 ± 12.87	54.54 ± 15.51	43.95 ± 22.56	50.46 ± 19.43	45.46 ± 20.90	44.90 ± 14.45	49.27 ± 22.09	**61.25 ± 13.30**
40984	61.43 ± 3.06	57.54 ± 6.25	62.59 ± 2.34	60.09 ± 3.36	**64.37 ± 3.76**	62.19 ± 2.44	59.57 ± 2.87	59.03 ± 4.29	56.17 ± 4.55	62.88 ± 2.80
46908	93.54 ± 5.23	90.76 ± 5.88	95.48 ± 4.84	92.12 ± 9.22	88.93 ± 12.94	84.66 ± 9.14	61.62 ± 24.48	80.16 ± 10.55	93.34 ± 6.47	**96.13 ± 2.89**
46911	51.80 ± 9.69	54.45 ± 12.53	56.20 ± 7.82	56.04 ± 12.64	47.47 ± 10.18	55.54 ± 9.75	54.19 ± 11.16	50.97 ± 5.27	54.43 ± 8.21	**58.17 ± 11.55**
46920	55.61 ± 8.97	57.72 ± 12.36	58.12 ± 12.36	60.64 ± 17.45	58.19 ± 13.56	59.83 ± 11.21	55.64 ± 11.06	60.07 ± 10.20	59.00 ± 4.50	**61.83 ± 10.46**
46924	48.73 ± 3.08	51.63 ± 4.47	51.75 ± 4.42	**54.60 ± 5.46**	48.28 ± 5.07	52.75 ± 8.98	52.30 ± 9.16	50.45 ± 2.84	49.75 ± 13.43	52.25 ± 11.49
46927	50.73 ± 14.82	48.75 ± 9.88	53.29 ± 8.08	54.00 ± 17.65	47.07 ± 13.00	52.32 ± 12.56	53.70 ± 14.63	48.87 ± 8.13	51.67 ± 9.31	**55.28 ± 10.90**
46929	**60.61 ± 14.55**	49.55 ± 25.85	54.34 ± 18.34	55.36 ± 14.13	55.98 ± 20.64	56.12 ± 23.24	56.29 ± 24.04	48.81 ± 27.65	54.98 ± 27.30	56.83 ± 20.49
46932	55.77 ± 5.01	56.97 ± 10.14	55.93 ± 5.04	58.57 ± 6.54	55.30 ± 7.29	52.54 ± 5.24	52.34 ± 4.77	**59.09 ± 7.52**	53.84 ± 7.49	56.37 ± 8.16
46937	51.35 ± 4.36	51.49 ± 3.72	52.70 ± 2.59	51.27 ± 3.17	51.66 ± 4.75	52.17 ± 3.80	49.61 ± 2.57	49.37 ± 1.88	50.34 ± 3.85	**53.85 ± 2.87**
46938	46.03 ± 14.35	45.21 ± 13.24	50.91 ± 13.68	48.78 ± 15.79	46.61 ± 13.82	49.86 ± 13.12	44.81 ± 15.80	54.90 ± 7.97	**56.14 ± 15.55**	48.10 ± 11.36
46941	40.35 ± 8.83	42.16 ± 7.88	42.77 ± 6.08	44.28 ± 4.77	40.89 ± 8.09	39.16 ± 7.26	**46.80 ± 10.24**	36.85 ± 7.89	39.16 ± 11.14	42.81 ± 7.15
46955	45.74 ± 8.53	36.84 ± 7.37	47.16 ± 6.28	40.51 ± 8.08	43.05 ± 5.59	43.87 ± 10.81	42.21 ± 11.94	41.03 ± 7.07	31.23 ± 6.19	**53.04 ± 12.30**
46956	56.23 ± 25.33	58.98 ± 20.97	63.16 ± 20.31	65.30 ± 22.89	58.16 ± 24.87	50.93 ± 23.57	**68.65 ± 18.28**	52.26 ± 13.75	63.52 ± 21.20	65.88 ± 20.06
46960	42.95 ± 5.78	40.13 ± 6.23	46.95 ± 4.41	46.55 ± 7.92	42.59 ± 6.61	46.81 ± 10.11	44.00 ± 11.21	36.10 ± 3.83	42.69 ± 7.11	**47.88 ± 8.26**
46963	54.39 ± 9.20	52.87 ± 12.82	57.91 ± 7.15	57.48 ± 8.74	51.40 ± 10.49	51.11 ± 11.11	43.54 ± 11.33	58.89 ± 9.22	51.73 ± 10.60	**64.65 ± 3.36**
46969	56.65 ± 16.61	61.34 ± 17.95	**65.85 ± 15.40**	61.69 ± 12.54	59.09 ± 12.20	61.91 ± 15.51	65.84 ± 16.28	64.38 ± 11.39	63.68 ± 15.21	63.01 ± 14.45
46979	44.03 ± 22.73	46.47 ± 20.57	46.86 ± 23.74	49.83 ± 24.42	45.53 ± 22.83	48.50 ± 21.07	49.47 ± 24.03	45.03 ± 19.85	46.91 ± 23.76	**50.09 ± 17.22**

Table 16: 5-shot results. Each entry is mean ± standard deviation over repeated runs; the best and second-best methods for each dataset are shown in bold and underlined, respectively.

OpenML ID	CatBoost	VIME	SAINT	SCARF	T-JEPA	STUNT	SubTab	SwitchTab	ICE-T	CCR
6	51.48 ± 2.22	37.93 ± 2.88	49.99 ± 3.10	35.21 ± 3.40	53.91 ± 3.08	47.24 ± 3.61	40.63 ± 2.92	44.18 ± 2.64	46.39 ± 2.20	**53.95 ± 1.33**
12	83.42 ± 4.23	77.58 ± 3.16	84.71 ± 2.77	76.92 ± 4.57	81.88 ± 2.91	86.38 ± 2.06	83.47 ± 2.17	**86.92 ± 1.48**	80.97 ± 2.58	85.88 ± 1.92
15	92.36 ± 3.60	94.57 ± 1.73	94.50 ± 1.75	94.14 ± 3.55	92.86 ± 4.04	94.71 ± 2.03	94.71 ± 1.36	93.57 ± 2.40	94.36 ± 2.27	**95.57 ± 0.95**
16	76.50 ± 3.56	62.81 ± 3.90	70.12 ± 3.22	73.67 ± 2.18	71.55 ± 4.11	77.78 ± 3.88	73.20 ± 3.16	**83.28 ± 1.46**	82.22 ± 1.64	81.28 ± 2.96
22	59.98 ± 2.24	57.01 ± 3.57	60.90 ± 2.82	54.38 ± 4.03	65.05 ± 2.67	62.90 ± 1.72	60.67 ± 2.64	**65.33 ± 2.54**	53.25 ± 4.18	61.95 ± 2.73
28	82.08 ± 2.38	81.71 ± 2.12	73.96 ± 2.93	83.69 ± 1.87	78.94 ± 4.48	79.60 ± 2.05	77.92 ± 0.86	**89.12 ± 1.62**	81.86 ± 2.10	87.09 ± 2.47
29	76.01 ± 4.45	74.57 ± 5.69	78.70 ± 3.16	78.67 ± 1.91	65.07 ± 7.18	77.25 ± 6.08	73.48 ± 6.14	60.29 ± 8.89	71.01 ± 7.35	**79.42 ± 3.48**
32	79.55 ± 1.62	78.36 ± 2.52	80.20 ± 3.55	81.02 ± 1.89	81.80 ± 1.53	82.82 ± 1.91	75.28 ± 2.97	82.95 ± 2.21	80.40 ± 2.46	**84.10 ± 2.35**
37	61.10 ± 6.05	62.01 ± 6.51	61.28 ± 5.00	63.08 ± 4.51	62.64 ± 4.93	63.38 ± 3.92	62.14 ± 5.77	54.42 ± 9.69	63.38 ± 4.64	**64.55 ± 5.24**
38	83.32 ± 14.50	64.60 ± 10.82	80.48 ± 6.01	76.60 ± 9.24	77.95 ± 7.06	73.72 ± 9.68	80.08 ± 9.84	57.66 ± 8.75	64.56 ± 14.49	**84.69 ± 4.88**
44	73.34 ± 5.39	64.12 ± 6.10	76.87 ± 3.69	79.74 ± 5.63	68.48 ± 10.27	76.29 ± 7.68	74.53 ± 5.87	74.56 ± 8.61	81.09 ± 5.72	**83.14 ± 4.39**
46	60.89 ± 8.02	55.00 ± 5.50	69.26 ± 9.68	78.62 ± 4.09	57.07 ± 12.05	**84.14 ± 5.62**	61.71 ± 8.03	45.27 ± 6.31	68.04 ± 10.55	84.01 ± 5.26
50	56.82 ± 8.73	55.59 ± 6.23	55.83 ± 7.70	57.08 ± 7.75	57.19 ± 9.32	58.65 ± 8.41	57.40 ± 6.09	54.43 ± 6.94	**75.94 ± 13.20**	68.85 ± 8.15
182	75.64 ± 4.06	76.30 ± 4.29	77.07 ± 1.68	76.91 ± 2.01	79.04 ± 3.07	79.28 ± 3.17	77.60 ± 2.60	77.12 ± 3.63	77.04 ± 3.69	**80.84 ± 1.69**
458	93.61 ± 2.51	86.63 ± 5.31	93.91 ± 4.23	95.62 ± 1.61	93.73 ± 1.75	97.75 ± 1.00	93.55 ± 1.68	**98.11 ± 0.87**	96.09 ± 1.61	97.99 ± 0.54
1067	64.76 ± 11.62	66.68 ± 9.00	67.37 ± 4.45	66.01 ± 5.99	66.75 ± 10.72	65.07 ± 6.33	65.73 ± 7.73	62.30 ± 11.43	65.81 ± 6.43	**68.10 ± 7.99**
1068	63.15 ± 12.48	64.83 ± 15.14	63.84 ± 12.86	64.34 ± 6.29	62.12 ± 12.42	65.08 ± 8.90	62.57 ± 7.89	63.83 ± 12.21	64.77 ± 11.74	**68.55 ± 8.53**
1468	65.65 ± 3.84	36.66 ± 3.96	OOM	53.06 ± 7.29	53.15 ± 3.96	47.13 ± 12.30	38.33 ± 3.80	53.06 ± 4.19	63.98 ± 5.32	**73.89 ± 3.97**
1478	75.40 ± 3.31	**76.10 ± 3.20**	OOM	50.72 ± 3.32	75.91 ± 4.79	68.61 ± 5.48	72.51 ± 2.86	53.64 ± 7.66	61.35 ± 10.80	75.94 ± 2.88
1480	59.91 ± 3.33	55.08 ± 7.33	57.78 ± 7.06	58.82 ± 4.15	54.53 ± 8.87	56.84 ± 7.29	52.99 ± 6.97	52.74 ± 6.31	57.61 ± 5.36	**60.60 ± 5.49**
1485	50.92 ± 4.11	50.22 ± 2.51	51.34 ± 2.71	53.09 ± 5.06	47.76 ± 1.06	52.37 ± 3.42	51.33 ± 4.38	51.27 ± 2.76	**53.98 ± 5.91**	52.46 ± 2.63
1489	65.77 ± 8.23	62.06 ± 6.38	67.32 ± 5.33	67.57 ± 3.65	68.18 ± 6.33	67.94 ± 5.28	69.50 ± 4.38	69.48 ± 6.81	65.46 ± 11.46	**69.79 ± 6.55**
1494	64.83 ± 8.92	63.40 ± 9.25	**69.05 ± 5.82**	67.81 ± 6.05	68.01 ± 5.87	66.32 ± 5.44	66.54 ± 4.57	66.73 ± 7.13	68.39 ± 8.12	66.50 ± 4.76
1501	67.37 ± 2.72	67.65 ± 3.51	64.58 ± 5.42	62.13 ± 4.89	68.35 ± 2.96	63.89 ± 4.22	60.28 ± 3.34	68.81 ± 4.17	**71.19 ± 3.63**	70.31 ± 2.44
4534	78.31 ± 10.63	71.64 ± 6.63	75.64 ± 4.02	81.17 ± 4.46	75.37 ± 8.50	77.08 ± 3.97	79.22 ± 5.01	76.68 ± 5.43	76.02 ± 9.68	**82.58 ± 5.03**
4538	33.10 ± 3.66	31.09 ± 3.61	33.44 ± 3.17	31.45 ± 2.66	32.09 ± 3.62	31.61 ± 4.00	29.31 ± 4.16	32.25 ± 4.32	29.75 ± 2.58	**33.66 ± 2.98**
6332	63.15 ± 6.19	63.72 ± 7.65	60.07 ± 5.03	57.03 ± 9.25	**64.74 ± 11.83**	61.67 ± 7.52	57.78 ± 4.08	51.20 ± 4.90	60.74 ± 9.16	61.11 ± 7.90
23381	52.20 ± 4.98	52.26 ± 5.49	51.26 ± 4.30	51.67 ± 5.67	52.50 ± 6.04	49.80 ± 6.53	49.70 ± 5.95	49.90 ± 3.75	52.30 ± 4.24	**56.12 ± 5.32**
40499	75.09 ± 2.85	78.47 ± 2.74	80.17 ± 1.65	64.62 ± 4.12	**86.03 ± 1.76**	77.47 ± 3.46	74.53 ± 2.43	78.59 ± 3.67	73.54 ± 2.53	83.25 ± 3.10
40668	**40.33 ± 3.93**	36.17 ± 7.55	37.41 ± 2.11	35.87 ± 3.33	37.09 ± 5.54	38.35 ± 3.93	36.48 ± 3.62	34.15 ± 4.73	38.44 ± 4.20	38.94 ± 4.72
40670	64.36 ± 9.30	59.55 ± 7.08	64.24 ± 6.48	75.78 ± 5.38	57.81 ± 9.49	78.21 ± 3.90	51.54 ± 5.46	49.06 ± 7.77	60.47 ± 6.58	**85.17 ± 3.75**
40979	83.03 ± 2.29	81.42 ± 3.18	37.03 ± 5.83	77.92 ± 5.89	78.65 ± 2.75	85.60 ± 3.34	79.00 ± 3.10	**85.65 ± 3.08**	83.22 ± 3.09	84.38 ± 1.67
40983	61.06 ± 11.41	61.19 ± 8.81	68.00 ± 16.89	60.13 ± 9.17	57.94 ± 11.95	71.15 ± 16.65	56.36 ± 6.96	51.91 ± 12.37	51.55 ± 14.88	**85.04 ± 6.62**
40984	76.77 ± 4.30	72.21 ± 3.00	76.79 ± 2.65	73.07 ± 3.28	**77.27 ± 3.60**	75.43 ± 2.35	74.59 ± 3.65	67.51 ± 2.86	69.70 ± 3.96	77.01 ± 4.26
46908	91.48 ± 4.07	90.88 ± 2.46	95.60 ± 2.87	92.37 ± 2.23	92.55 ± 2.26	92.29 ± 6.13	85.13 ± 14.94	83.83 ± 9.40	92.05 ± 3.87	**96.25 ± 1.13**
46911	56.35 ± 9.27	54.70 ± 7.75	56.32 ± 9.17	56.16 ± 8.13	51.65 ± 6.90	51.83 ± 5.05	58.97 ± 5.49	50.56 ± 7.21	56.17 ± 7.07	**65.41 ± 6.37**
46920	66.13 ± 4.21	67.59 ± 7.70	**73.53 ± 4.71**	70.25 ± 9.05	72.85 ± 6.67	71.11 ± 6.85	71.57 ± 5.62	69.95 ± 7.47	68.37 ± 5.10	71.39 ± 7.48
46924	54.89 ± 5.84	51.75 ± 2.13	55.83 ± 4.52	54.72 ± 5.90	52.16 ± 2.85	55.23 ± 5.73	56.49 ± 7.33	52.42 ± 3.62	**61.99 ± 4.31**	57.16 ± 8.00
46927	**63.50 ± 10.27**	50.39 ± 6.19	60.13 ± 9.39	60.29 ± 6.65	55.04 ± 7.59	56.99 ± 6.51	61.17 ± 7.67	49.73 ± 5.13	49.23 ± 6.41	57.88 ± 7.34
46929	61.10 ± 16.60	59.14 ± 10.14	54.46 ± 11.56	61.00 ± 13.96	60.46 ± 8.89	60.20 ± 11.06	50.52 ± 12.25	58.05 ± 11.00	62.79 ± 12.93	**65.63 ± 13.49**
46932	**61.86 ± 7.05**	58.55 ± 7.92	59.75 ± 6.05	58.69 ± 8.05	61.65 ± 7.37	57.32 ± 5.15	57.25 ± 4.58	61.57 ± 6.41	60.66 ± 7.26	59.97 ± 6.27
46937	52.62 ± 6.08	52.30 ± 4.80	**52.82 ± 5.47**	51.39 ± 2.61	52.35 ± 6.18	52.09 ± 3.82	51.74 ± 3.56	48.87 ± 4.23	50.08 ± 2.87	52.17 ± 4.42
46938	49.28 ± 6.67	49.46 ± 8.60	53.03 ± 10.15	48.90 ± 9.72	51.39 ± 8.72	57.19 ± 6.46	53.30 ± 6.71	**57.22 ± 9.83**	51.16 ± 11.04	53.09 ± 8.27
46941	50.74 ± 3.26	46.76 ± 8.42	52.78 ± 3.70	48.77 ± 3.68	53.05 ± 6.85	47.44 ± 6.33	52.02 ± 5.08	44.68 ± 7.41	51.28 ± 7.55	**53.07 ± 4.97**
46955	**69.02 ± 6.18**	40.05 ± 3.38	63.55 ± 4.67	53.28 ± 7.09	61.68 ± 7.12	53.43 ± 6.32	48.72 ± 6.83	48.04 ± 6.09	36.13 ± 3.86	64.14 ± 8.86
46956	64.51 ± 10.31	59.10 ± 14.02	63.41 ± 10.89	65.42 ± 9.37	58.24 ± 12.15	62.26 ± 10.04	65.28 ± 9.64	58.63 ± 12.26	65.82 ± 13.72	**66.13 ± 8.54**
46960	**55.12 ± 6.01**	43.03 ± 7.32	51.39 ± 6.61	50.99 ± 5.54	48.44 ± 8.99	50.87 ± 7.66	47.60 ± 6.07	39.56 ± 4.01	48.95 ± 6.10	51.82 ± 4.14
46963	60.37 ± 7.51	56.71 ± 6.75	58.03 ± 7.94	60.62 ± 8.45	59.13 ± 8.55	54.98 ± 10.07	54.35 ± 7.62	60.37 ± 8.93	59.93 ± 8.12	**62.20 ± 5.25**
46969	71.01 ± 11.85	75.22 ± 7.95	80.54 ± 4.50	75.14 ± 10.48	69.66 ± 14.00	78.79 ± 10.29	**81.55 ± 9.02**	76.38 ± 13.43	77.95 ± 8.98	80.91 ± 6.10
46979	59.53 ± 9.62	54.49 ± 10.03	59.07 ± 7.87	58.01 ± 8.14	56.02 ± 6.13	57.72 ± 7.72	58.60 ± 9.48	**60.41 ± 10.23**	56.17 ± 8.39	59.87 ± 9.33

Table 17: 10-shot results. Each entry is mean ± standard deviation over repeated runs; the best and second-best methods for each dataset are shown in bold and underlined, respectively.

OpenML ID	CatBoost	VIME	SAINT	SCARF	T-JEPA	STUNT	SubTab	SwitchTab	ICE-T	CCR
6	62.14 ± 1.67	46.76 ± 2.91	63.70 ± 0.96	44.71 ± 2.43	64.95 ± 1.45	58.88 ± 1.25	51.24 ± 1.08	53.25 ± 2.24	57.93 ± 2.03	**66.31 ± 0.40**
12	89.22 ± 1.96	83.58 ± 2.03	91.38 ± 1.40	81.70 ± 2.76	91.15 ± 2.73	89.65 ± 1.65	87.83 ± 0.94	**91.75 ± 1.25**	86.58 ± 1.19	89.40 ± 1.50
15	94.14 ± 0.94	**95.86 ± 1.20**	94.79 ± 1.01	95.43 ± 0.91	95.57 ± 0.93	94.64 ± 1.40	95.85 ± 1.15	93.93 ± 1.52	95.43 ± 0.96	95.71 ± 0.64
16	84.28 ± 1.89	72.11 ± 3.10	83.03 ± 2.98	76.88 ± 2.18	84.53 ± 2.17	83.20 ± 1.84	80.72 ± 2.41	**88.10 ± 0.95**	85.83 ± 1.69	87.95 ± 1.51
22	67.67 ± 1.17	63.30 ± 2.56	66.92 ± 2.36	58.62 ± 3.12	**72.12 ± 2.43**	69.47 ± 2.53	66.30 ± 1.47	69.17 ± 2.47	59.92 ± 2.18	70.42 ± 2.36
28	89.48 ± 1.81	87.65 ± 1.50	87.44 ± 1.86	87.76 ± 1.45	88.38 ± 2.12	86.39 ± 2.14	85.85 ± 1.75	**94.03 ± 1.45**	90.10 ± 1.74	92.05 ± 1.24
29	76.04 ± 5.02	76.59 ± 3.36	78.91 ± 2.21	78.79 ± 4.01	70.58 ± 4.12	76.96 ± 8.91	77.61 ± 5.11	61.23 ± 5.16	75.36 ± 5.23	**79.13 ± 1.83**
32	83.40 ± 2.12	83.84 ± 2.06	85.68 ± 1.64	84.11 ± 1.78	86.89 ± 1.47	87.21 ± 1.57	82.91 ± 1.43	86.52 ± 1.67	86.33 ± 1.68	**89.52 ± 1.26**
37	61.49 ± 5.87	63.25 ± 5.65	62.00 ± 6.87	64.10 ± 4.30	62.83 ± 6.76	60.65 ± 8.11	60.91 ± 5.77	55.19 ± 4.97	62.73 ± 5.57	**67.14 ± 3.09**
38	86.41 ± 6.67	71.35 ± 6.69	84.60 ± 6.64	83.19 ± 4.10	86.10 ± 8.41	79.09 ± 7.22	82.11 ± 4.18	69.18 ± 8.17	77.34 ± 4.24	**88.42 ± 4.23**
44	77.50 ± 4.21	67.09 ± 4.98	77.75 ± 2.26	**85.65 ± 2.23**	72.21 ± 5.60	75.49 ± 9.38	74.67 ± 5.60	78.18 ± 3.70	85.45 ± 2.49	84.61 ± 2.21
46	77.02 ± 5.19	67.97 ± 5.46	82.48 ± 6.73	82.59 ± 3.61	70.64 ± 6.81	85.52 ± 4.09	66.80 ± 5.06	55.11 ± 4.45	73.04 ± 8.14	**86.55 ± 2.50**
50	57.55 ± 5.18	56.07 ± 5.56	57.14 ± 5.77	57.76 ± 4.49	55.52 ± 8.19	58.80 ± 6.08	51.51 ± 6.40	51.88 ± 4.15	**82.34 ± 6.13**	70.68 ± 7.25
182	81.14 ± 1.94	80.52 ± 2.56	82.52 ± 1.76	79.74 ± 1.38	81.94 ± 1.40	82.36 ± 1.79	81.35 ± 2.35	80.74 ± 2.03	81.17 ± 2.33	**82.55 ± 1.58**
458	93.67 ± 2.57	91.54 ± 3.08	94.38 ± 3.12	96.45 ± 0.84	93.91 ± 3.55	97.99 ± 1.16	94.20 ± 1.78	**98.64 ± 0.49**	97.28 ± 1.28	98.05 ± 0.38
1067	65.26 ± 8.67	66.80 ± 10.15	68.09 ± 7.60	68.13 ± 9.64	68.67 ± 10.96	62.37 ± 11.63	60.73 ± 9.01	63.36 ± 9.08	64.19 ± 6.73	**70.05 ± 6.90**
1068	61.98 ± 11.21	64.95 ± 12.15	66.96 ± 9.32	67.38 ± 5.79	68.15 ± 8.95	65.36 ± 9.07	67.88 ± 10.31	61.26 ± 14.09	61.58 ± 10.60	**68.81 ± 3.43**
1468	75.83 ± 3.58	48.42 ± 3.75	OOM	59.81 ± 6.43	68.56 ± 5.11	52.27 ± 11.37	51.85 ± 3.29	66.81 ± 4.30	68.89 ± 3.70	**80.79 ± 3.74**
1478	80.62 ± 2.68	80.76 ± 1.95	OOM	53.57 ± 1.81	82.34 ± 2.45	75.95 ± 2.16	79.67 ± 2.08	56.15 ± 9.03	65.38 ± 8.60	**85.77 ± 1.76**
1480	58.63 ± 6.73	57.91 ± 6.90	59.32 ± 4.41	**62.39 ± 4.10**	60.68 ± 6.93	60.26 ± 5.20	60.26 ± 7.91	55.04 ± 7.95	59.40 ± 5.21	60.77 ± 5.08
1485	50.25 ± 1.64	50.46 ± 2.06	51.46 ± 1.69	53.40 ± 3.96	50.96 ± 1.02	50.50 ± 3.71	50.62 ± 2.98	50.52 ± 2.55	54.21 ± 4.87	**55.04 ± 3.01**
1489	69.37 ± 4.83	62.18 ± 6.77	70.54 ± 3.77	69.97 ± 3.15	69.23 ± 5.43	68.77 ± 5.08	69.34 ± 4.05	69.13 ± 3.87	69.79 ± 4.54	**71.01 ± 4.66**
1494	72.42 ± 5.53	69.74 ± 3.99	71.41 ± 4.17	74.95 ± 4.14	71.75 ± 7.13	72.21 ± 5.98	68.86 ± 6.50	67.39 ± 4.43	71.61 ± 4.73	**75.36 ± 4.88**
1501	74.80 ± 2.26	75.13 ± 2.00	77.56 ± 2.39	67.49 ± 5.29	78.90 ± 3.53	74.20 ± 3.54	70.25 ± 2.47	**80.09 ± 2.37**	80.00 ± 1.81	78.06 ± 0.84
4534	85.24 ± 9.02	77.27 ± 4.67	77.96 ± 2.98	85.31 ± 4.20	79.06 ± 10.28	76.40 ± 5.81	80.47 ± 3.58	78.76 ± 6.95	77.31 ± 5.86	**86.23 ± 4.16**
4538	**37.62 ± 2.66**	34.36 ± 2.85	33.65 ± 2.86	33.91 ± 2.55	35.75 ± 2.42	33.53 ± 2.57	31.75 ± 1.89	32.49 ± 1.35	32.20 ± 3.05	35.15 ± 2.49
6332	57.50 ± 8.69	**66.12 ± 7.90**	60.79 ± 6.79	57.15 ± 7.69	64.56 ± 2.93	61.94 ± 8.12	57.04 ± 9.00	49.63 ± 7.98	62.78 ± 7.26	65.93 ± 4.77
23381	54.80 ± 4.44	52.62 ± 5.11	55.24 ± 4.66	51.79 ± 4.65	53.00 ± 5.33	48.00 ± 5.87	52.50 ± 5.06	51.50 ± 5.10	51.30 ± 5.83	**56.74 ± 3.99**
40499	80.97 ± 1.36	83.38 ± 1.98	86.62 ± 1.42	69.84 ± 2.49	**91.98 ± 1.38**	83.70 ± 2.65	80.55 ± 2.35	82.34 ± 3.57	79.16 ± 1.63	89.04 ± 1.75
40668	**39.78 ± 5.05**	36.29 ± 6.04	38.87 ± 1.98	35.99 ± 6.47	37.42 ± 3.62	36.10 ± 2.31	38.44 ± 3.43	36.38 ± 4.58	35.96 ± 4.22	39.41 ± 2.25
40670	74.91 ± 3.69	69.19 ± 5.10	71.65 ± 5.88	78.64 ± 4.91	66.82 ± 9.32	80.22 ± 3.90	58.54 ± 4.36	57.49 ± 6.23	73.89 ± 2.03	**87.43 ± 2.94**
40979	88.72 ± 1.50	89.35 ± 1.55	45.46 ± 5.24	81.78 ± 5.30	88.00 ± 1.36	88.95 ± 1.89	84.33 ± 1.87	**91.17 ± 1.29**	89.38 ± 1.58	89.17 ± 1.37
40983	70.97 ± 8.28	61.44 ± 10.02	68.12 ± 15.53	66.12 ± 5.36	67.09 ± 9.73	74.27 ± 11.53	63.17 ± 8.98	61.04 ± 9.69	59.81 ± 10.32	**92.79 ± 4.03**
40984	81.69 ± 3.34	76.23 ± 2.42	82.02 ± 1.54	75.48 ± 1.41	81.58 ± 2.62	78.31 ± 2.95	77.58 ± 1.94	69.70 ± 3.57	73.51 ± 2.81	**83.38 ± 1.67**
46908	92.62 ± 4.41	91.00 ± 4.76	95.72 ± 4.16	92.62 ± 5.45	93.55 ± 1.96	93.15 ± 6.07	89.17 ± 8.66	88.18 ± 8.83	91.24 ± 7.37	**96.37 ± 2.79**
46911	60.54 ± 6.18	55.03 ± 5.27	58.06 ± 7.87	58.69 ± 6.85	63.16 ± 5.81	58.20 ± 5.97	59.02 ± 8.66	54.28 ± 3.95	56.97 ± 5.07	**67.82 ± 5.50**
46920	65.48 ± 6.31	69.32 ± 8.61	74.65 ± 5.77	72.21 ± 3.34	74.73 ± 5.56	73.05 ± 6.12	70.47 ± 5.92	74.68 ± 6.56	70.66 ± 5.98	**76.18 ± 3.47**
46924	62.67 ± 2.15	51.87 ± 2.45	58.54 ± 3.65	60.40 ± 2.69	55.67 ± 4.50	57.70 ± 5.16	59.60 ± 3.84	51.26 ± 3.86	**65.25 ± 2.53**	62.82 ± 4.07
46927	63.13 ± 8.55	50.51 ± 3.17	62.84 ± 8.03	63.37 ± 9.40	58.43 ± 8.96	59.23 ± 6.86	**63.70 ± 7.80**	50.87 ± 3.65	49.70 ± 4.79	63.48 ± 6.34
46929	**67.52 ± 9.59**	59.26 ± 9.67	60.83 ± 10.30	61.12 ± 10.18	62.10 ± 7.33	61.07 ± 9.34	58.19 ± 8.40	55.29 ± 4.58	66.24 ± 12.72	65.13 ± 9.84
46932	62.65 ± 3.36	60.57 ± 3.84	61.85 ± 3.62	60.60 ± 4.53	60.72 ± 3.53	57.79 ± 3.95	58.15 ± 4.06	62.48 ± 3.39	60.96 ± 3.14	**64.26 ± 2.63**
46937	**55.28 ± 5.35**	51.82 ± 1.69	53.63 ± 3.34	51.51 ± 2.12	54.12 ± 4.21	51.03 ± 3.15	52.13 ± 3.37	52.22 ± 4.31	51.84 ± 2.67	54.06 ± 2.53
46938	56.78 ± 5.13	53.42 ± 10.20	53.15 ± 6.82	52.88 ± 10.17	56.02 ± 10.31	**60.58 ± 6.78**	56.20 ± 9.47	59.51 ± 12.97	53.42 ± 8.79	58.01 ± 6.95
46941	55.22 ± 4.49	54.50 ± 5.71	58.59 ± 6.64	54.96 ± 4.40	**59.31 ± 7.48**	53.69 ± 5.27	55.17 ± 7.21	47.83 ± 4.72	56.11 ± 6.51	58.81 ± 5.82
46955	**80.18 ± 3.21**	42.49 ± 2.40	74.67 ± 4.09	61.19 ± 7.03	78.17 ± 5.85	60.67 ± 5.55	52.83 ± 4.81	56.28 ± 4.01	38.91 ± 3.71	74.08 ± 3.24
46956	60.35 ± 7.48	59.22 ± 5.75	63.66 ± 5.37	65.54 ± 7.81	61.25 ± 8.33	56.73 ± 8.97	59.15 ± 5.45	55.32 ± 6.45	57.18 ± 4.86	**66.38 ± 8.28**
46960	**58.35 ± 6.76**	48.90 ± 5.24	56.58 ± 5.80	52.92 ± 5.58	51.90 ± 7.49	57.16 ± 5.32	52.11 ± 5.46	43.60 ± 5.75	53.15 ± 5.24	56.98 ± 5.96
46963	71.03 ± 3.35	66.82 ± 5.11	69.86 ± 4.49	67.34 ± 2.83	70.74 ± 5.33	68.49 ± 6.35	66.64 ± 6.27	67.42 ± 3.92	70.41 ± 7.38	**71.64 ± 2.69**
46969	79.79 ± 8.96	82.74 ± 2.81	85.24 ± 3.42	84.83 ± 4.42	75.31 ± 7.87	87.93 ± 2.24	**88.42 ± 2.35**	84.90 ± 3.31	83.29 ± 5.22	88.19 ± 2.00
46979	59.19 ± 10.39	55.48 ± 6.03	59.19 ± 11.62	58.26 ± 10.91	52.41 ± 8.11	58.35 ± 8.56	56.18 ± 10.92	53.26 ± 8.63	54.18 ± 6.23	**59.99 ± 7.42**

Table 18: 1-shot Macro-F1 results. Each entry is mean ± standard deviation over repeated runs; the best and second-best methods for each dataset are shown in bold and underlined, respectively.

OpenML ID	CatBoost	VIME	SAINT	SCARF	T-JEPA	STUNT	SubTab	SwitchTab	ICE-T	CCR
6	25.83 ± 3.70	19.64 ± 2.47	23.85 ± 2.30	18.93 ± 2.70	25.89 ± 3.03	25.08 ± 3.28	21.37 ± 2.87	23.90 ± 3.76	23.43 ± 2.55	**26.32 ± 3.89**
12	60.39 ± 4.75	53.64 ± 7.43	62.22 ± 6.54	61.11 ± 9.56	61.23 ± 8.13	**69.98 ± 8.83**	64.38 ± 7.25	66.16 ± 6.74	61.60 ± 6.16	63.85 ± 7.32
15	77.76 ± 18.64	89.97 ± 7.88	92.31 ± 1.24	90.32 ± 8.05	87.08 ± 9.84	93.06 ± 1.79	91.21 ± 6.18	92.97 ± 3.84	**94.25 ± 1.25**	91.34 ± 6.15
16	39.73 ± 7.75	39.19 ± 5.10	39.15 ± 8.29	55.56 ± 5.67	47.49 ± 4.80	62.19 ± 7.55	49.93 ± 8.25	**62.80 ± 6.44**	62.33 ± 6.61	60.03 ± 7.85
22	33.36 ± 4.16	38.47 ± 6.09	37.26 ± 3.52	34.74 ± 4.27	42.33 ± 4.87	41.84 ± 4.66	40.29 ± 5.48	**42.35 ± 4.77**	32.47 ± 3.11	39.20 ± 4.22
28	55.31 ± 6.10	59.77 ± 5.53	53.51 ± 4.91	**71.61 ± 3.69**	58.08 ± 4.35	63.47 ± 5.33	54.60 ± 4.11	69.11 ± 6.01	61.73 ± 5.40	70.40 ± 4.04
29	56.93 ± 12.06	54.96 ± 8.80	58.68 ± 12.11	57.70 ± 17.06	55.12 ± 10.23	60.85 ± 11.72	50.88 ± 8.37	49.10 ± 5.47	51.36 ± 12.26	**62.05 ± 12.02**
32	50.56 ± 3.94	56.13 ± 6.88	55.76 ± 4.23	57.69 ± 3.44	56.40 ± 4.73	**61.38 ± 4.13**	51.50 ± 4.37	59.72 ± 5.62	53.97 ± 5.65	60.51 ± 3.59
37	54.27 ± 8.45	54.56 ± 11.93	54.63 ± 6.40	**58.54 ± 9.40**	53.62 ± 9.66	53.58 ± 8.40	57.82 ± 4.01	48.33 ± 12.58	54.29 ± 9.94	55.83 ± 9.00
38	36.92 ± 18.38	41.91 ± 12.04	53.91 ± 8.54	51.09 ± 5.08	25.99 ± 12.86	42.99 ± 8.18	**54.18 ± 9.24**	32.78 ± 9.50	37.17 ± 17.86	51.31 ± 17.14
44	72.51 ± 8.68	57.48 ± 11.43	70.55 ± 5.70	76.62 ± 10.99	66.99 ± 4.58	67.56 ± 9.27	69.00 ± 8.77	77.54 ± 6.91	78.16 ± 7.09	**79.36 ± 9.57**
46	39.69 ± 3.86	40.73 ± 7.05	39.41 ± 6.43	63.03 ± 7.87	40.54 ± 6.05	55.34 ± 19.25	48.14 ± 6.31	38.51 ± 4.34	48.93 ± 9.95	**70.55 ± 10.99**
50	48.21 ± 6.12	49.14 ± 6.67	49.95 ± 5.32	48.91 ± 6.49	48.91 ± 6.49	46.78 ± 5.73	46.89 ± 3.91	47.57 ± 4.63	**54.06 ± 11.54**	50.79 ± 7.47
182	58.58 ± 7.81	59.12 ± 10.13	56.05 ± 7.27	62.00 ± 10.21	59.46 ± 8.41	61.91 ± 8.19	60.34 ± 9.01	63.10 ± 10.04	56.92 ± 6.47	**64.63 ± 9.24**
458	70.49 ± 9.12	70.45 ± 9.46	80.62 ± 7.12	**92.25 ± 3.64**	78.86 ± 10.12	85.98 ± 10.27	85.30 ± 7.86	91.52 ± 4.62	91.88 ± 5.00	89.10 ± 6.76
1067	51.49 ± 18.02	54.20 ± 16.99	52.44 ± 16.23	50.53 ± 16.03	52.43 ± 16.74	51.46 ± 16.31	50.54 ± 16.77	51.94 ± 18.74	50.81 ± 18.05	**57.01 ± 7.62**
1068	47.84 ± 12.23	45.60 ± 13.71	48.81 ± 11.59	44.49 ± 14.09	45.75 ± 13.98	47.55 ± 13.11	47.87 ± 12.25	**50.61 ± 12.96**	47.30 ± 12.86	45.69 ± 14.24
1468	43.47 ± 4.27	16.31 ± 5.07	OOM	33.29 ± 8.41	24.55 ± 5.81	29.43 ± 15.04	17.91 ± 3.28	32.16 ± 7.04	**46.79 ± 6.13**	39.96 ± 6.36
1478	44.62 ± 7.59	55.33 ± 7.94	OOM	39.80 ± 6.83	51.12 ± 8.20	49.44 ± 7.97	53.34 ± 6.06	42.73 ± 7.68	42.94 ± 11.10	**55.83 ± 6.68**
1480	47.74 ± 9.60	45.21 ± 7.91	46.52 ± 10.96	**51.47 ± 6.29**	45.68 ± 8.47	48.19 ± 8.55	41.14 ± 7.69	48.39 ± 12.69	48.12 ± 10.31	49.39 ± 8.62
1485	47.41 ± 1.87	47.92 ± 4.34	48.34 ± 1.38	48.62 ± 6.25	48.48 ± 2.26	48.18 ± 3.70	46.60 ± 3.81	49.56 ± 3.61	48.01 ± 10.12	**49.82 ± 8.26**
1489	58.51 ± 8.37	53.49 ± 7.29	59.33 ± 5.45	**61.62 ± 4.70**	56.96 ± 7.53	55.94 ± 10.54	56.60 ± 10.95	58.29 ± 13.39	54.80 ± 15.06	59.54 ± 9.34
1494	55.96 ± 8.06	52.99 ± 13.11	57.92 ± 6.90	56.86 ± 5.63	53.35 ± 8.99	57.43 ± 8.37	51.65 ± 8.20	56.61 ± 8.47	**60.64 ± 6.78**	57.98 ± 6.79
1501	39.60 ± 4.14	43.21 ± 6.07	40.46 ± 5.13	42.14 ± 7.37	25.43 ± 2.31	45.02 ± 4.48	40.82 ± 4.96	41.75 ± 4.50	47.05 ± 4.40	**48.08 ± 6.31**
4534	55.99 ± 9.74	59.38 ± 10.19	54.99 ± 8.27	62.41 ± 14.13	57.56 ± 9.84	59.56 ± 12.00	59.18 ± 13.48	63.59 ± 13.31	60.11 ± 14.02	**65.43 ± 18.22**
4538	22.34 ± 4.48	21.91 ± 5.29	21.34 ± 4.51	**24.18 ± 5.69**	23.59 ± 5.18	23.73 ± 5.59	23.30 ± 4.58	22.63 ± 5.71	23.15 ± 5.89	22.45 ± 4.25
6332	54.21 ± 9.52	53.71 ± 9.80	48.92 ± 10.86	47.22 ± 10.55	48.19 ± 7.11	51.21 ± 10.95	49.14 ± 11.48	48.49 ± 4.68	54.58 ± 9.91	**54.59 ± 7.47**
23381	48.22 ± 7.12	48.57 ± 5.65	48.89 ± 3.94	48.60 ± 6.24	50.98 ± 6.90	43.28 ± 7.17	45.54 ± 5.50	50.50 ± 7.09	49.83 ± 8.52	**51.79 ± 5.13**
40499	54.07 ± 5.46	59.04 ± 6.22	57.85 ± 3.45	47.63 ± 2.29	62.64 ± 5.55	61.53 ± 3.70	53.88 ± 3.89	**64.66 ± 4.62**	56.54 ± 3.74	63.84 ± 3.81
40668	29.77 ± 2.53	29.95 ± 4.05	28.56 ± 3.60	26.92 ± 2.88	30.10 ± 2.73	29.19 ± 2.64	29.07 ± 5.20	29.28 ± 2.54	29.12 ± 3.52	**30.47 ± 2.18**
40670	40.79 ± 6.04	43.98 ± 8.41	44.61 ± 9.17	57.27 ± 8.34	37.50 ± 3.62	47.70 ± 15.65	39.98 ± 7.49	41.30 ± 3.58	51.42 ± 6.75	**58.47 ± 12.24**
40979	53.34 ± 7.11	59.31 ± 7.20	22.12 ± 2.96	58.98 ± 8.37	56.27 ± 6.58	**66.02 ± 8.79**	55.99 ± 6.14	56.34 ± 6.69	59.94 ± 6.36	60.91 ± 5.25
40983	37.33 ± 8.73	36.02 ± 11.88	39.36 ± 6.16	39.94 ± 8.27	33.16 ± 12.79	38.74 ± 11.80	33.96 ± 11.58	35.15 ± 9.49	35.77 ± 11.26	**46.54 ± 15.45**
40984	59.56 ± 3.37	55.16 ± 6.99	59.49 ± 2.27	58.12 ± 4.03	**61.69 ± 4.86**	59.93 ± 3.81	57.01 ± 3.26	56.99 ± 4.66	55.33 ± 5.31	60.69 ± 3.25
46908	64.61 ± 8.88	61.17 ± 7.24	62.80 ± 4.17	65.66 ± 8.72	62.25 ± 12.94	63.35 ± 9.63	44.47 ± 16.42	52.65 ± 6.74	67.62 ± 8.51	**69.93 ± 3.37**
46911	46.99 ± 7.68	47.95 ± 9.20	49.87 ± 5.86	48.21 ± 7.07	44.53 ± 8.24	47.03 ± 6.85	46.64 ± 6.82	46.47 ± 3.49	47.16 ± 6.16	**50.96 ± 10.83**
46920	54.20 ± 9.48	55.47 ± 12.61	57.89 ± 12.10	58.44 ± 16.17	57.06 ± 13.48	58.01 ± 11.84	52.22 ± 11.51	58.54 ± 10.21	55.34 ± 5.81	**59.74 ± 11.68**
46924	48.35 ± 2.97	47.71 ± 3.52	47.78 ± 4.00	**51.58 ± 5.23**	45.96 ± 4.48	51.08 ± 8.32	50.48 ± 9.55	48.37 ± 1.74	48.96 ± 13.33	48.98 ± 11.34
46927	45.17 ± 10.00	44.12 ± 6.91	45.91 ± 6.64	46.67 ± 12.17	42.36 ± 13.00	46.83 ± 9.08	46.83 ± 9.56	44.09 ± 4.91	46.25 ± 4.82	**48.08 ± 6.01**
46929	**44.10 ± 7.89**	35.55 ± 15.24	40.32 ± 8.61	37.43 ± 7.80	40.30 ± 20.64	40.77 ± 13.54	39.55 ± 12.46	34.97 ± 15.36	37.50 ± 16.64	38.70 ± 22.47
46932	50.86 ± 7.75	52.97 ± 12.07	50.93 ± 7.34	54.74 ± 9.00	50.79 ± 9.09	49.50 ± 6.96	47.96 ± 7.03	**57.50 ± 9.63**	50.21 ± 9.70	52.13 ± 11.37
46937	48.38 ± 2.52	49.37 ± 3.17	**51.87 ± 2.53**	49.09 ± 2.32	47.62 ± 5.90	50.59 ± 5.98	47.48 ± 2.45	47.55 ± 1.57	47.78 ± 4.03	48.82 ± 3.26
46938	34.30 ± 7.69	34.16 ± 8.21	43.46 ± 5.42	39.83 ± 8.87	34.77 ± 7.21	39.62 ± 7.38	34.23 ± 9.32	**43.91 ± 6.51**	41.85 ± 7.21	35.50 ± 9.04
46941	40.45 ± 10.51	37.99 ± 6.79	41.88 ± 8.24	40.08 ± 6.48	38.84 ± 8.88	37.12 ± 8.38	**43.74 ± 10.04**	34.78 ± 8.41	38.73 ± 12.54	40.87 ± 10.17
46955	42.29 ± 8.34	32.71 ± 5.12	44.18 ± 5.75	36.30 ± 6.72	31.85 ± 3.04	38.84 ± 8.87	39.91 ± 11.08	36.29 ± 4.84	28.13 ± 4.52	**46.20 ± 10.98**
46956	40.93 ± 14.61	42.98 ± 11.60	45.18 ± 12.00	47.10 ± 12.41	41.27 ± 13.34	39.07 ± 15.52	47.74 ± 9.38	38.53 ± 7.55	46.62 ± 12.79	**48.81 ± 12.01**
46960	39.99 ± 3.84	36.24 ± 5.47	40.06 ± 4.37	40.13 ± 5.51	37.43 ± 3.32	**43.67 ± 9.22**	38.79 ± 8.26	33.65 ± 3.57	38.92 ± 5.67	41.33 ± 7.02
46963	45.21 ± 6.85	44.58 ± 10.55	48.86 ± 5.19	47.62 ± 5.89	43.64 ± 6.56	42.98 ± 8.83	38.45 ± 8.35	49.48 ± 7.38	44.52 ± 7.77	**52.56 ± 2.79**
46969	51.09 ± 18.31	55.37 ± 18.48	61.19 ± 16.41	54.43 ± 12.85	51.81 ± 12.25	59.59 ± 14.93	**63.11 ± 16.59**	62.79 ± 11.48	60.24 ± 15.69	53.01 ± 16.27
46979	37.15 ± 14.73	36.84 ± 10.56	37.69 ± 14.29	**39.93 ± 14.64**	36.75 ± 22.83	39.86 ± 13.88	38.62 ± 14.14	36.61 ± 13.47	36.68 ± 15.76	37.95 ± 14.60

Table 19: 5-shot Macro-F1 results. Each entry is mean ± standard deviation over repeated runs; the best and second-best methods for each dataset are shown in bold and underlined, respectively.

OpenML ID	CatBoost	VIME	SAINT	SCARF	T-JEPA	STUNT	SubTab	SwitchTab	ICE-T	CCR
6	50.91 ± 2.18	38.11 ± 2.84	50.15 ± 3.04	35.39 ± 3.18	54.07 ± 3.10	47.27 ± 3.53	40.79 ± 2.77	44.50 ± 2.49	46.67 ± 2.04	**54.28 ± 1.37**
12	83.44 ± 4.37	77.37 ± 3.29	84.62 ± 2.93	76.68 ± 4.74	81.78 ± 2.88	86.51 ± 2.04	83.61 ± 2.13	**86.97 ± 1.48**	81.11 ± 2.54	85.77 ± 2.02
15	91.18 ± 4.58	93.94 ± 2.10	93.87 ± 2.07	93.21 ± 4.57	91.62 ± 5.21	94.08 ± 2.46	94.09 ± 1.69	92.84 ± 2.82	93.65 ± 2.86	**95.10 ± 1.06**
16	76.41 ± 3.58	62.44 ± 3.99	69.98 ± 3.20	73.68 ± 2.11	71.30 ± 3.89	77.66 ± 4.05	73.13 ± 3.29	**83.26 ± 1.56**	82.33 ± 1.58	81.12 ± 3.01
22	59.41 ± 2.42	56.56 ± 3.57	60.76 ± 2.66	54.10 ± 4.00	**64.74 ± 2.56**	62.09 ± 1.89	60.40 ± 2.56	64.31 ± 2.38	53.39 ± 4.12	61.53 ± 3.08
28	82.06 ± 2.31	81.51 ± 2.22	74.09 ± 3.00	83.70 ± 1.95	78.98 ± 4.53	79.33 ± 2.02	77.85 ± 0.89	**89.07 ± 1.73**	82.00 ± 2.11	87.09 ± 2.46
29	74.94 ± 6.40	74.12 ± 5.96	78.30 ± 3.13	78.15 ± 1.97	64.12 ± 7.79	76.87 ± 6.07	73.11 ± 6.10	59.16 ± 8.64	70.60 ± 7.28	**78.92 ± 3.40**
32	78.99 ± 1.81	77.88 ± 2.71	79.79 ± 3.77	80.75 ± 2.03	81.42 ± 1.65	82.49 ± 2.13	74.96 ± 2.94	82.77 ± 2.31	79.96 ± 2.75	**83.72 ± 2.61**
37	58.39 ± 5.84	59.28 ± 7.58	58.94 ± 5.72	61.52 ± 5.30	60.65 ± 6.12	60.89 ± 4.35	60.08 ± 4.39	51.53 ± 8.51	60.13 ± 5.83	**62.64 ± 5.29**
38	**67.69 ± 14.71**	48.17 ± 6.54	65.87 ± 8.16	59.78 ± 10.21	48.95 ± 10.18	56.32 ± 6.48	62.17 ± 8.34	44.46 ± 5.43	49.73 ± 9.79	66.47 ± 4.85
44	71.22 ± 5.39	63.17 ± 6.23	75.10 ± 2.95	79.13 ± 5.21	66.67 ± 10.25	75.65 ± 7.72	73.52 ± 5.14	74.01 ± 8.57	80.30 ± 5.54	**82.02 ± 4.65**
46	59.96 ± 7.25	53.38 ± 5.27	68.22 ± 8.55	77.58 ± 4.13	56.19 ± 11.48	**83.36 ± 5.31**	60.55 ± 7.40	43.55 ± 5.38	66.14 ± 9.62	83.14 ± 5.15
50	53.55 ± 8.40	52.91 ± 5.21	52.45 ± 6.64	53.82 ± 6.36	54.02 ± 8.45	55.29 ± 7.91	52.93 ± 5.65	52.17 ± 6.07	**73.49 ± 14.27**	66.15 ± 7.51
182	73.77 ± 3.65	74.67 ± 4.10	75.55 ± 2.34	74.03 ± 2.24	76.51 ± 3.22	76.62 ± 3.38	75.42 ± 3.13	74.29 ± 3.23	75.01 ± 3.62	**78.31 ± 2.54**
458	93.41 ± 2.97	84.63 ± 5.96	93.96 ± 3.14	95.55 ± 1.67	94.41 ± 1.37	97.89 ± 1.03	94.59 ± 1.56	**98.00 ± 1.08**	96.09 ± 1.37	97.95 ± 0.71
1067	55.07 ± 9.34	56.88 ± 7.31	57.28 ± 4.18	55.93 ± 5.13	53.85 ± 7.67	55.85 ± 4.72	56.27 ± 4.72	53.08 ± 8.76	56.03 ± 5.06	**57.40 ± 4.60**
1068	48.36 ± 5.64	48.13 ± 9.08	48.93 ± 7.95	49.32 ± 2.64	47.74 ± 7.32	49.29 ± 4.98	47.85 ± 4.49	46.87 ± 5.99	47.50 ± 6.27	**50.38 ± 5.31**
1468	68.34 ± 3.39	36.52 ± 4.10	OOM	51.81 ± 7.48	53.89 ± 4.23	46.80 ± 12.54	38.19 ± 3.81	53.28 ± 4.01	63.82 ± 5.62	**73.49 ± 4.00**
1478	74.79 ± 3.44	**76.00 ± 3.55**	OOM	50.50 ± 3.67	75.25 ± 5.12	68.12 ± 5.52	72.06 ± 2.92	52.15 ± 7.67	60.94 ± 10.38	75.80 ± 2.74
1480	54.86 ± 4.74	50.88 ± 5.54	54.70 ± 6.90	52.78 ± 5.56	51.45 ± 9.03	53.25 ± 7.31	50.83 ± 6.09	49.66 ± 6.41	55.18 ± 5.77	**57.27 ± 5.16**
1485	48.90 ± 4.07	48.69 ± 3.20	49.49 ± 2.48	50.92 ± 5.54	47.33 ± 0.76	51.42 ± 4.28	48.83 ± 4.46	50.65 ± 2.82	**52.83 ± 6.14**	50.10 ± 2.47
1489	63.48 ± 7.36	58.44 ± 6.82	65.53 ± 4.73	64.69 ± 3.54	64.68 ± 5.58	65.46 ± 4.01	**66.87 ± 4.03**	64.62 ± 5.65	62.91 ± 10.08	65.89 ± 6.15
1494	61.98 ± 7.86	61.08 ± 9.05	**68.34 ± 4.54**	67.47 ± 5.77	66.71 ± 5.40	62.44 ± 5.05	63.92 ± 5.24	64.11 ± 7.01	66.20 ± 7.78	63.18 ± 4.83
1501	67.06 ± 2.92	67.32 ± 3.66	63.73 ± 5.62	61.72 ± 5.11	68.35 ± 2.96	63.34 ± 4.34	59.77 ± 3.61	68.49 ± 4.35	**71.27 ± 3.66**	70.32 ± 2.41
4534	78.06 ± 10.61	71.21 ± 6.78	75.13 ± 3.81	81.12 ± 4.45	75.12 ± 8.49	76.84 ± 3.96	79.00 ± 4.99	76.36 ± 5.78	75.46 ± 9.91	**82.32 ± 5.05**
4538	30.36 ± 2.35	29.21 ± 2.78	31.24 ± 2.37	29.89 ± 2.77	30.24 ± 3.32	29.81 ± 3.87	27.74 ± 3.52	30.07 ± 3.58	27.81 ± 2.79	**31.46 ± 2.58**
6332	61.92 ± 5.71	**62.73 ± 7.49**	59.28 ± 4.53	56.23 ± 8.83	53.87 ± 10.95	60.53 ± 6.91	56.42 ± 3.06	49.82 ± 3.57	59.55 ± 9.20	59.71 ± 7.85
23381	49.33 ± 4.68	49.82 ± 4.95	49.01 ± 3.96	48.72 ± 5.19	51.65 ± 6.23	48.07 ± 5.98	48.94 ± 5.47	49.00 ± 2.92	51.09 ± 3.94	**51.91 ± 4.86**
40499	74.78 ± 2.55	77.99 ± 2.86	79.65 ± 1.47	63.85 ± 4.09	**85.86 ± 1.84**	77.05 ± 3.66	74.04 ± 2.60	78.34 ± 3.84	73.22 ± 2.34	83.01 ± 3.35
40668	**33.46 ± 2.48**	30.07 ± 4.42	31.86 ± 0.48	29.91 ± 1.83	31.54 ± 2.80	31.67 ± 1.94	31.80 ± 2.77	29.96 ± 2.49	30.88 ± 2.13	32.08 ± 2.34
40670	62.23 ± 8.72	57.64 ± 6.97	61.73 ± 6.14	74.55 ± 5.41	56.42 ± 9.14	76.53 ± 4.15	50.04 ± 5.11	46.61 ± 7.39	58.61 ± 5.80	**83.89 ± 4.28**
40979	82.96 ± 2.41	81.40 ± 3.21	36.26 ± 5.58	77.77 ± 6.17	78.54 ± 2.79	**85.64 ± 3.35**	79.20 ± 2.99	**85.64 ± 3.07**	83.33 ± 3.07	84.27 ± 1.87
40983	45.54 ± 6.61	43.66 ± 3.70	51.03 ± 9.97	43.75 ± 4.37	43.82 ± 7.12	55.06 ± 14.01	42.08 ± 3.43	39.51 ± 7.15	39.57 ± 8.41	**67.21 ± 8.10**
40984	76.18 ± 4.38	71.12 ± 3.11	76.18 ± 2.65	72.41 ± 3.60	**77.00 ± 3.55**	75.07 ± 2.33	73.96 ± 3.78	66.55 ± 2.95	68.90 ± 3.71	76.54 ± 4.35
46908	62.38 ± 5.64	61.29 ± 2.88	66.92 ± 2.28	65.91 ± 4.93	64.78 ± 2.26	65.95 ± 5.74	59.15 ± 11.66	55.35 ± 6.75	63.73 ± 5.43	**70.18 ± 3.26**
46911	50.56 ± 7.24	48.07 ± 5.98	50.11 ± 6.38	48.33 ± 6.92	47.47 ± 5.05	46.78 ± 3.24	52.30 ± 3.48	45.45 ± 4.35	50.84 ± 5.71	**58.92 ± 5.67**
46920	65.23 ± 4.36	66.70 ± 8.58	**72.05 ± 4.41**	69.37 ± 9.86	70.66 ± 6.67	68.33 ± 6.31	70.95 ± 6.00	69.20 ± 8.43	67.28 ± 5.61	69.53 ± 7.99
46924	52.35 ± 5.86	49.79 ± 1.72	54.63 ± 4.45	51.70 ± 5.77	51.00 ± 2.54	53.82 ± 5.29	55.14 ± 7.39	50.30 ± 2.88	**61.03 ± 4.40**	55.48 ± 6.62
46927	**60.87 ± 9.17**	47.83 ± 4.50	59.76 ± 9.01	58.63 ± 7.12	49.53 ± 7.59	53.31 ± 6.78	57.92 ± 6.04	47.64 ± 3.86	47.44 ± 4.13	53.74 ± 8.54
46929	47.15 ± 10.21	44.12 ± 5.35	40.91 ± 6.72	45.77 ± 9.40	43.53 ± 8.89	45.20 ± 7.21	38.98 ± 6.86	42.87 ± 5.76	47.31 ± 8.10	**48.35 ± 7.98**
46932	**61.63 ± 7.22**	58.11 ± 7.90	59.02 ± 6.59	57.87 ± 8.81	61.15 ± 7.74	56.78 ± 5.33	56.56 ± 5.42	61.03 ± 6.46	60.12 ± 7.62	59.23 ± 6.88
46937	51.37 ± 5.89	49.49 ± 5.07	**52.74 ± 5.65**	47.04 ± 2.34	51.57 ± 6.08	50.78 ± 3.73	51.20 ± 3.32	47.62 ± 3.22	47.11 ± 2.68	48.31 ± 4.41
46938	41.33 ± 3.62	44.30 ± 4.56	43.60 ± 5.55	40.52 ± 7.10	43.03 ± 5.03	46.39 ± 4.34	44.73 ± 4.74	**46.72 ± 5.69**	42.17 ± 7.19	44.50 ± 5.69
46941	51.00 ± 3.40	47.31 ± 7.98	52.55 ± 3.68	48.18 ± 5.25	51.78 ± 6.83	47.80 ± 6.25	52.20 ± 4.93	44.50 ± 7.78	51.43 ± 6.87	**52.98 ± 5.66**
46955	**62.80 ± 6.18**	37.78 ± 2.93	57.83 ± 4.67	56.35 ± 5.97	57.98 ± 7.97	52.10 ± 6.89	49.67 ± 6.04	47.63 ± 4.77	33.45 ± 1.83	58.74 ± 8.13
46956	47.26 ± 5.45	43.10 ± 7.66	47.39 ± 5.67	47.57 ± 5.84	43.48 ± 6.86	46.46 ± 5.10	47.94 ± 5.52	43.83 ± 7.02	49.12 ± 8.15	**49.71 ± 4.97**
46960	**51.92 ± 5.87**	40.82 ± 5.74	48.86 ± 5.92	47.58 ± 5.07	45.79 ± 8.76	48.54 ± 6.51	46.06 ± 5.08	38.13 ± 3.81	46.63 ± 5.35	49.08 ± 3.98
46963	54.32 ± 5.04	50.57 ± 5.64	51.00 ± 5.46	52.79 ± 6.37	49.41 ± 6.76	48.44 ± 7.30	48.84 ± 6.37	52.71 ± 5.44	52.94 ± 7.20	**54.59 ± 3.62**
46969	70.42 ± 11.93	74.61 ± 7.86	80.45 ± 4.15	73.59 ± 11.03	68.25 ± 13.52	78.00 ± 9.93	**80.78 ± 8.75**	75.42 ± 13.44	77.04 ± 8.53	80.74 ± 5.71
46979	**52.88 ± 6.48**	48.11 ± 5.81	51.67 ± 4.30	50.46 ± 5.26	47.61 ± 6.13	50.02 ± 5.74	50.80 ± 5.72	51.51 ± 6.41	49.39 ± 5.32	52.02 ± 4.72

Table 20: 10-shot Macro-F1 results. Each entry is mean ± standard deviation over repeated runs; the best and second-best methods for each dataset are shown in bold and underlined, respectively.

OpenML ID	CatBoost	VIME	SAINT	SCARF	T-JEPA	STUNT	SubTab	SwitchTab	ICE-T	CCR
6	62.08 ± 1.76	46.80 ± 2.89	64.08 ± 1.04	44.82 ± 2.35	66.10 ± 1.45	58.90 ± 1.12	51.24 ± 1.05	53.43 ± 2.00	58.36 ± 1.95	**66.48 ± 0.30**
12	89.22 ± 1.99	83.58 ± 2.09	91.37 ± 1.45	81.66 ± 2.89	91.12 ± 2.78	89.70 ± 1.62	87.84 ± 0.98	**91.79 ± 1.23**	86.64 ± 1.16	89.41 ± 1.48
15	93.47 ± 1.11	**95.48 ± 1.30**	94.26 ± 1.08	94.95 ± 1.01	95.14 ± 0.97	94.15 ± 1.52	95.45 ± 1.25	93.29 ± 1.73	94.96 ± 1.14	95.30 ± 0.67
16	84.24 ± 1.89	71.96 ± 3.16	82.95 ± 3.02	76.82 ± 2.18	84.43 ± 2.21	83.07 ± 1.84	80.69 ± 2.46	**88.11 ± 0.97**	85.87 ± 1.71	87.91 ± 1.53
22	67.08 ± 1.29	62.90 ± 2.52	66.72 ± 2.20	58.14 ± 2.85	**71.93 ± 2.23**	68.61 ± 2.89	65.68 ± 1.23	68.55 ± 2.25	59.72 ± 1.82	70.11 ± 1.81
28	89.42 ± 1.85	87.63 ± 1.54	87.46 ± 1.83	87.71 ± 1.51	88.34 ± 2.15	86.18 ± 2.25	85.83 ± 1.76	**94.02 ± 1.49**	90.12 ± 1.74	92.02 ± 1.26
29	75.87 ± 4.96	76.20 ± 3.46	78.67 ± 2.36	78.27 ± 3.97	70.33 ± 3.97	76.55 ± 8.85	77.11 ± 5.13	60.18 ± 5.77	75.07 ± 5.14	**78.78 ± 1.68**
32	82.96 ± 2.37	83.69 ± 2.14	85.36 ± 1.84	83.99 ± 1.90	86.67 ± 1.52	87.02 ± 1.69	82.65 ± 1.37	86.37 ± 1.86	86.08 ± 1.76	**89.46 ± 1.27**
37	59.25 ± 5.76	61.30 ± 5.82	59.06 ± 6.19	61.64 ± 4.87	61.68 ± 7.17	58.70 ± 8.41	59.34 ± 4.98	53.46 ± 5.39	61.44 ± 5.19	**65.24 ± 3.14**
38	**72.26 ± 8.19**	53.80 ± 4.67	67.41 ± 6.87	64.37 ± 4.02	48.02 ± 5.59	60.72 ± 6.01	62.49 ± 4.49	52.26 ± 5.27	58.53 ± 3.16	71.93 ± 5.60
44	76.18 ± 5.05	66.33 ± 5.04	76.67 ± 2.39	**85.21 ± 2.18**	71.52 ± 5.44	74.57 ± 9.61	74.06 ± 5.51	77.67 ± 3.52	84.82 ± 2.58	83.85 ± 2.35
46	76.07 ± 5.20	66.86 ± 4.95	81.08 ± 6.64	81.69 ± 3.40	69.25 ± 7.14	84.76 ± 3.66	65.95 ± 4.88	52.87 ± 4.31	71.87 ± 7.74	**85.86 ± 2.39**
50	56.62 ± 4.67	54.39 ± 5.15	55.69 ± 5.03	56.36 ± 4.42	53.17 ± 6.19	56.43 ± 4.82	49.71 ± 6.36	50.93 ± 3.58	**81.00 ± 6.20**	68.59 ± 6.39
182	79.23 ± 1.72	78.06 ± 2.30	80.73 ± 1.39	77.53 ± 1.27	79.58 ± 1.33	80.68 ± 1.57	79.55 ± 1.89	78.25 ± 1.76	79.28 ± 2.01	**81.00 ± 1.27**
458	94.08 ± 2.68	90.54 ± 3.92	94.19 ± 2.68	96.26 ± 0.90	94.84 ± 3.04	96.72 ± 1.25	95.12 ± 1.36	**98.21 ± 0.54**	97.15 ± 1.12	97.92 ± 0.44
1067	55.77 ± 5.96	56.93 ± 7.76	57.40 ± 4.36	58.00 ± 5.97	52.24 ± 8.98	53.11 ± 7.24	53.31 ± 6.39	54.51 ± 5.76	55.16 ± 4.68	**59.87 ± 6.32**
1068	47.52 ± 6.58	48.25 ± 7.88	49.59 ± 5.97	**51.70 ± 3.52**	51.55 ± 5.67	49.75 ± 5.71	51.42 ± 6.27	47.27 ± 8.53	46.73 ± 6.06	50.50 ± 2.65
1468	76.87 ± 3.29	48.80 ± 3.83	OOM	58.82 ± 6.39	68.71 ± 5.07	51.97 ± 11.70	51.63 ± 3.05	66.50 ± 4.71	68.60 ± 4.08	**80.43 ± 3.80**
1478	80.31 ± 2.66	80.39 ± 2.02	OOM	53.57 ± 1.90	82.00 ± 2.43	75.81 ± 2.28	79.52 ± 1.95	55.53 ± 8.95	65.21 ± 8.50	**85.60 ± 1.73**
1480	56.08 ± 6.50	53.68 ± 5.35	56.42 ± 4.44	**57.76 ± 3.50**	56.36 ± 6.14	56.68 ± 4.34	56.77 ± 6.41	52.36 ± 6.14	56.80 ± 4.42	57.56 ± 4.98
1485	49.48 ± 2.27	48.81 ± 3.60	49.61 ± 1.98	51.91 ± 5.82	49.68 ± 2.43	49.92 ± 3.94	49.83 ± 3.18	49.42 ± 2.80	53.66 ± 4.75	**54.54 ± 2.87**
1489	66.95 ± 4.67	58.56 ± 6.16	67.66 ± 3.71	66.09 ± 3.62	65.12 ± 5.26	66.32 ± 4.27	65.31 ± 3.99	65.33 ± 4.17	66.02 ± 4.24	**68.42 ± 4.45**
1494	71.07 ± 5.47	67.89 ± 4.35	69.80 ± 4.12	72.36 ± 3.63	70.54 ± 6.82	72.07 ± 5.32	68.09 ± 6.14	65.72 ± 4.05	70.24 ± 4.26	**73.56 ± 5.44**
1501	74.85 ± 2.33	77.13 ± 2.01	77.46 ± 2.44	67.26 ± 5.59	78.90 ± 3.53	73.95 ± 3.72	69.80 ± 2.80	80.02 ± 2.50	**80.07 ± 1.82**	78.17 ± 0.73
4534	84.85 ± 9.49	76.99 ± 4.62	77.48 ± 3.06	85.18 ± 4.15	78.76 ± 10.20	75.90 ± 6.20	80.28 ± 3.51	78.53 ± 6.95	77.13 ± 5.93	**86.01 ± 4.40**
4538	**35.55 ± 2.32**	31.10 ± 2.16	31.74 ± 2.25	31.03 ± 2.24	34.21 ± 2.42	31.99 ± 2.58	30.57 ± 1.55	30.78 ± 1.42	30.56 ± 2.79	32.30 ± 2.11
6332	56.36 ± 8.66	**65.15 ± 8.10**	59.91 ± 6.78	56.35 ± 7.97	52.85 ± 2.97	61.40 ± 7.99	56.44 ± 9.20	48.79 ± 7.60	62.23 ± 7.24	64.96 ± 4.81
23381	50.83 ± 3.86	49.98 ± 5.03	51.04 ± 4.29	49.11 ± 4.77	51.76 ± 5.81	46.63 ± 5.46	51.10 ± 4.33	50.59 ± 4.51	50.84 ± 5.53	**52.03 ± 5.56**
40499	80.85 ± 1.29	83.22 ± 2.01	86.47 ± 1.40	69.22 ± 2.66	**92.01 ± 1.38**	83.55 ± 2.75	80.39 ± 2.45	82.10 ± 3.85	78.95 ± 1.82	89.01 ± 1.79
40668	**34.42 ± 3.07**	30.66 ± 3.84	34.00 ± 1.22	30.03 ± 3.56	32.83 ± 1.80	31.74 ± 1.67	33.50 ± 2.00	31.52 ± 2.51	30.90 ± 2.26	32.20 ± 0.81
40670	73.41 ± 3.57	67.61 ± 4.83	70.03 ± 5.82	77.45 ± 4.54	65.01 ± 8.52	78.43 ± 4.47	57.62 ± 3.86	54.08 ± 4.58	71.45 ± 2.37	**86.12 ± 3.00**
40979	88.73 ± 1.50	89.30 ± 1.59	45.05 ± 5.27	81.75 ± 5.47	87.98 ± 1.32	89.03 ± 1.84	84.31 ± 1.87	**91.18 ± 1.24**	89.42 ± 1.56	89.14 ± 1.41
40983	52.52 ± 5.69	43.78 ± 5.07	51.15 ± 9.75	48.76 ± 3.39	50.72 ± 6.34	55.48 ± 8.27	46.52 ± 4.46	43.98 ± 4.33	44.92 ± 6.13	**78.31 ± 6.34**
40984	81.04 ± 3.41	75.49 ± 2.44	81.06 ± 1.81	74.80 ± 1.36	81.27 ± 2.65	77.84 ± 2.98	76.81 ± 2.15	69.03 ± 3.28	72.57 ± 3.13	**83.09 ± 1.66**
46908	65.22 ± 6.19	62.62 ± 6.81	67.00 ± 3.90	66.16 ± 6.65	65.48 ± 1.96	66.45 ± 6.20	62.41 ± 9.41	59.95 ± 7.08	64.60 ± 8.11	**70.43 ± 4.84**
46911	54.85 ± 4.79	49.16 ± 3.93	53.09 ± 5.12	53.58 ± 5.45	51.39 ± 3.94	51.96 ± 4.50	52.84 ± 5.66	49.33 ± 2.69	52.30 ± 3.39	**61.72 ± 3.75**
46920	64.89 ± 7.10	68.12 ± 9.79	74.17 ± 6.20	71.60 ± 3.35	72.48 ± 5.56	72.10 ± 6.02	70.17 ± 5.83	74.22 ± 7.39	70.18 ± 6.20	**75.63 ± 4.24**
46924	60.71 ± 2.10	49.91 ± 2.86	57.60 ± 3.68	59.22 ± 3.29	54.43 ± 4.85	57.04 ± 4.87	58.96 ± 3.53	50.62 ± 3.69	**64.66 ± 2.98**	61.91 ± 4.28
46927	60.21 ± 7.47	47.95 ± 2.49	60.00 ± 6.95	59.71 ± 7.74	52.58 ± 8.96	55.97 ± 6.06	**60.61 ± 6.10**	48.79 ± 3.04	47.82 ± 4.17	60.39 ± 5.28
46929	48.57 ± 5.57	44.24 ± 5.21	45.27 ± 5.52	47.89 ± 5.71	44.71 ± 7.33	47.61 ± 5.70	43.42 ± 4.43	41.92 ± 2.54	49.88 ± 7.75	**51.39 ± 3.17**
46932	62.42 ± 3.61	59.54 ± 4.22	61.14 ± 3.84	60.19 ± 4.48	60.43 ± 3.63	57.29 ± 4.30	57.88 ± 4.20	62.34 ± 3.37	60.65 ± 3.10	**63.77 ± 2.82**
46937	**54.87 ± 5.27**	50.20 ± 2.69	51.69 ± 3.33	49.33 ± 1.48	52.91 ± 4.38	50.47 ± 3.03	51.45 ± 3.11	50.47 ± 3.95	51.11 ± 2.23	52.89 ± 2.40
46938	44.82 ± 3.62	44.42 ± 7.42	43.72 ± 4.34	43.05 ± 6.63	46.22 ± 5.81	47.91 ± 3.45	46.64 ± 6.15	**48.41 ± 8.19**	44.30 ± 4.69	48.22 ± 7.46
46941	56.02 ± 4.80	53.80 ± 6.29	58.95 ± 6.50	55.27 ± 4.52	**59.34 ± 7.20**	54.21 ± 4.84	55.22 ± 7.09	47.97 ± 4.65	56.60 ± 6.16	59.17 ± 5.59
46955	**74.56 ± 3.21**	41.51 ± 2.45	69.42 ± 4.09	57.64 ± 6.00	74.21 ± 5.84	57.75 ± 5.10	52.77 ± 3.91	54.93 ± 3.38	35.87 ± 2.40	69.33 ± 4.64
46956	45.98 ± 4.55	43.22 ± 3.82	47.51 ± 2.62	**47.69 ± 4.31**	43.12 ± 5.34	43.35 ± 4.89	44.97 ± 3.38	42.11 ± 3.37	44.33 ± 2.84	46.73 ± 3.52
46960	**54.24 ± 4.91**	45.49 ± 4.57	53.28 ± 4.80	50.12 ± 4.92	48.71 ± 6.81	53.74 ± 4.05	49.17 ± 5.05	41.00 ± 4.93	50.53 ± 3.89	53.92 ± 4.76
46963	62.89 ± 3.11	57.27 ± 4.23	62.71 ± 3.93	59.17 ± 2.78	62.06 ± 4.74	59.97 ± 6.21	58.16 ± 5.45	58.24 ± 3.85	61.69 ± 6.59	**63.07 ± 2.00**
46969	79.05 ± 8.61	80.79 ± 3.22	84.03 ± 3.63	83.78 ± 4.44	74.49 ± 7.72	86.95 ± 2.45	**87.25 ± 2.70**	83.77 ± 3.35	82.22 ± 5.18	87.02 ± 2.41
46979	51.92 ± 7.21	48.23 ± 4.86	50.50 ± 7.26	50.58 ± 7.03	48.54 ± 8.11	50.85 ± 5.88	49.56 ± 6.76	47.97 ± 6.06	48.60 ± 4.41	**52.14 ± 6.65**