BRIDGED CLUSTERING: SEMI-SUPERVISED SPARSE BRIDGING

Anonymous authors
Paper under double-blind review

ABSTRACT

We introduce **BRIDGED CLUSTERING**, a semi-supervised framework to learn predictors from any *unpaired* input $\mathcal{X}$ and output $\mathcal{Y}$ dataset. Our method first clusters $\mathcal{X}$ and $\mathcal{Y}$ independently, then learns a sparse, interpretable bridge between the $\mathcal{X}$ and $\mathcal{Y}$ clusters using only a few paired examples. At inference, a new input x is assigned to its nearest input cluster, and the centroid of the linked output cluster is returned as the prediction $\hat{y}$. Unlike traditional semi-supervised learning, BRIDGED CLUSTERING *explicitly* leverages output-only data, and unlike dense transport-based methods, it maintains a sparse and interpretable alignment. Through theoretical analysis, we show that with bounded mis-clustering and mis-bridging rates, our algorithm becomes an effective and efficient predictor. Empirically, our method is competitive with SOTA methods while remaining simple, model-agnostic, and label-efficient in low-supervision settings.

1 INTRODUCTION

Modern data analytics often involve reconciling rich but unaligned data collected for different purposes. For example, researchers may study a large morphological collection of museum specimen records alongside an independent genetic sequence repository such as GenBank Sayers et al. (2019), without reliably coordinated specimen identifiers to join them Mulcahy (2022). This scenario is common across domains such as biomedicine, ecology, and the social sciences, where modern data ecosystems rarely provide perfectly paired records for every variable of interest Putrama & Martinek (2024); Alma'aitah et al. (2024). Building supervised models in these settings often requires costly annotation, which becomes challenging at scale.

When labeled pairs are scarce, *semi-supervised learning* (SSL) is a natural approach, leveraging large pools of unlabeled inputs alongside limited supervised samples. Standard SSL assumes that each unlabeled input $x \in \mathcal{X}$ has a hidden label $y \in \mathcal{Y}$ Van Engelen & Hoos (2020); Yang et al. (2023), but this is often violated in multimodal settings, where datasets may include *output-only data* with no corresponding x. Examples for output-only data include: unpaired DNA barcodes collected independently of morphological images Ratnasingham et al. (2024), clinical outcome records separate from patient biosignals Mark (2016), satellite imagery without matching weather station logs, audio recordings lacking textual transcriptions, or microscopy images without corresponding gene-sequencing profiles. In these settings, fully exploiting the available data requires learning not only from unlabeled inputs, but also from the distributional structure and abundance of unlabeled outputs. Motivated by this challenge, we ask:

> *Can SSL be extended to incorporate rich, unlinked datasets of inputs and outputs for effective prediction, through simple clustering techniques?*

Key challenges. Extending SSL to unlinked inputs and outputs raises several challenges. First, SSL objectives rely on unlabeled inputs and instance-level pairing, and therefore cannot be directly applied when outputs are observed independently of inputs. Second, while output-only data encode valuable structural information about $\mathcal{Y}$, this information is not grounded in $\mathcal{X}$, making it unclear how it should influence prediction. Third, approaches that align inputs and outputs only at the marginal or distributional level risk coarse solutions. Finally, any practical method must not introduce substantial computational overhead or strong modeling assumptions.

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.

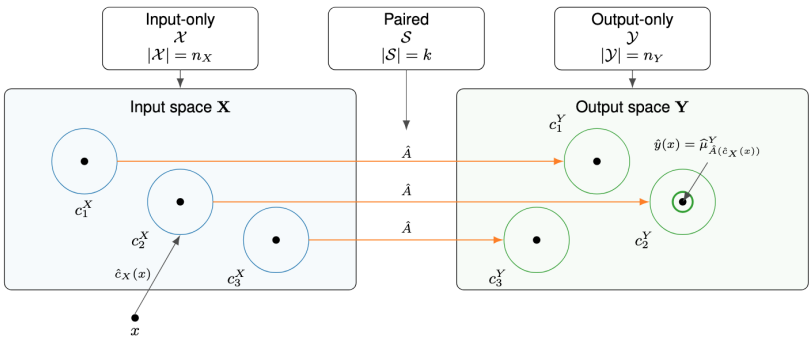


Figure 1: Bridged Clustering overview: (1) Cluster $\mathcal{X}$ and $\mathcal{Y}$ independently, (2) learn a sparse cluster bridge $\hat{A}$ from k paired samples by majority vote, (3) predict by mapping x to its input cluster and returning the centroid of the bridged output cluster.

Several lines of work partially address the challenges above. Transport-based alignment methods exploit output-only data by matching marginal distributions across spaces, either via cross-domain costs or by aligning intra-space geometries Mémoli (2011); Cuturi (2013); Peyré & Cuturi (2019). However, these approaches operate at the distributional level and may yield degenerate or overly coarse alignments. Related work on unmatched regression leverages separate samples of inputs and outputs under restrictive structural assumptions Carpentier & Schlüter (2016); Rigollet & Weed (2019); Azadkia & Balabdaoui (2024). Finally, generative and multi-view methods can effectively impute missing modalities, but typically require strong parametric assumptions or complete instance-level pairing Dempster et al. (1977); Blum & Mitchell (1998); Bickel & Scheffer (2004).

We address these challenges at the *cluster level*: we independently exploit unpaired data to structure $\mathcal{X}$ and $\mathcal{Y}$, and use a small paired set to learn a sparse, interpretable bridge between them. Unlike pretrained multimodal models that rely on large-scale external paired data (e.g., CLIP, 2021), we ask whether cross-modal prediction can instead be learned transparently from the available unpaired data and a few observed pairs, without external data or pretraining.

BRIDGED CLUSTERING is a model-agnostic SSL framework for learning from unpaired inputs and outputs with minimal paired supervision. It clusters each space independently, learns a sparse cluster-level association from a few paired examples, and predicts by mapping inputs to their linked output clusters; exploiting structure in both modalities reduces predictive risk as unpaired data increase. We evaluate BRIDGED CLUSTERING across vision, language, and bioinformatics domains. Our contributions are:

- We introduce a **cluster-level SSL framework** that learns from unpaired input-only and output-only data using minimal paired supervision.
- We show how **independently collected heterogeneous datasets** can be unified through sparse cluster-level bridges, enabling effective prediction without instance-level alignment.
- We present a **simple, interpretable, efficient, and model-agnostic** algorithm that scales linearly with the number of examples and provides transparent cluster correspondences.

Our results point to a complementary direction for SSL: **jointly exploiting input-only and output-only data through independent structure discovery**.

2 RELATED WORK

Semi-Supervised and Self-Supervised Learning: By leveraging large sets of unlabeled inputs, SSL methods such as graph-based label propagation Zhu & Ghahramani (2002); Zhou et al. (2003), Laplacian Regularized Least Squares (LapRLS; Belkin et al. (2006)), consistency regularization Laine & Aila (2016); Tarvainen & Valpola (2017), and hybrid approaches such as MixMatch Berthelot et al. (2019) and FixMatch Sohn et al. (2020) can substantially reduce annotation costs. Recent

regression-specific methods include Twin Neural Network Regression (TNNR) Wetzel et al. (2022), Uncertainty-Consistent Variational Model Ensembling (UCVME) Dai et al. (2023), and RankUp Huang et al. (2024). Unlike BRIDGED CLUSTERING, these methods use unlabeled inputs but not output-only data: their objectives do not jointly exploit unlabeled inputs and additional outputs, even when both are abundant.

Self-supervised contrastive methods like SimCLR Chen et al. (2020) involve unlabeled multi-view augmentations, and deep clustering methods such as SCAN Van Gansbeke et al. (2020) and NNM Dang et al. (2021) also rely on multi-view consistency. These approaches use multiple augmented views, but operate within a single input modality, while our model jointly targets inputs and outputs.

Multi-View Learning: Multi-view and co-training methods learn from multiple feature spaces, making them conceptually close to our setting. Related work includes classical co-training and PAC-style extensions Blum & Mitchell (1998); Dasgupta et al. (2001); Balcan et al. (2004); Balcan & Blum (2010), dependency-seeking clustering Rey & Roth (2012), mixture-based models such as Bayesian local CCA mixtures Viinikanoja et al. (2010), and canonical least-squares clustering Lei et al. (2017). However, these methods assume paired views (X, Y) for each instance, whereas BRIDGED CLUSTERING learns a mapping from X to Y when most instances are unpaired. They therefore cannot exploit the disjoint $\mathcal{X}$- and $\mathcal{Y}$-only corpora available to BRIDGED CLUSTERING.

Regression with Unmatched Samples: Recent statistical work on unmatched regression addresses disjoint $\mathcal{X}$ and $\mathcal{Y}$ data, typically under strong structural assumptions. Azadkia & Balabdaoui (2024) formulate linear regression as a deconvolution problem, while others assume monotone or isotonic structure Carpentier & Schlüter (2016); Rigollet & Weed (2019). Generative approaches likewise assume latent parametric structure, such as mixture-of-regression models optimized via EM Dempster et al. (1977). In contrast, BRIDGED CLUSTERING makes none of these assumptions while solving the same problem of aligning disjoint inputs and outputs.

Distributional alignment: Optimal transport (OT) aligns input-output domains via a cross-space cost $c(x, y)$ and can predict through *barycentric mapping* Cuturi (2013); Peyré & Cuturi (2019). When such a cost is unavailable, *Gromov-Wasserstein* (GW) instead matches intra-space distances to recover a soft correspondence Mémoli (2011); Peyré et al. (2016). Both can exploit output-only data but may yield coarse alignments and incur substantial computational costs at scale Peyré & Cuturi (2019). In contrast, BRIDGED CLUSTERING uses a small paired set to learn a sparse, interpretable cluster-level bridge, requiring only clustering and majority voting (Appendix D).

Cross-Domain Mapping without Parallel Data: In NLP, unsupervised machine translation aligns monolingual embeddings and uses back-translation to bridge languages Lample et al. (2017). In computer vision, CycleGAN learns bidirectional domain mappings via cycle consistency Zhu et al. (2017); Huang et al. (2018). Unlike these end-to-end generative approaches, which rely on reconstructive losses in both directions, BRIDGED CLUSTERING is model-agnostic and learns a lightweight cluster-level association from few labeled pairs.

3 NOTATION

In this section, we introduce our notation. The input features are denoted by random variable $X \in \mathbb{R}^d$ and the output features are denoted by random variable $Y \in \mathbb{R}^{d'}$.

We assume that X and Y are mixtures of C latent groups. While the criteria defining these groups are unknown, we assume the existence of C distinct categories. Let

$$T \in [C] := \{1, \ldots, C\}$$

be a latent categorical variable drawn with probabilities $\pi_1, \ldots, \pi_C$. For each group t, let P_t^X and P_t^Y denote the distributions over X and Y. Conditional on $T = t$, we observe a pair:

$$(X, Y) \,\big|\, T = t \;\sim\; P_t^X \times P_t^Y,$$

and let $\mu_t^X := \mathbb{E}[X \mid T = t]$ and $\mu_t^Y := \mathbb{E}[Y \mid T = t]$.

Our method takes as input the following training data:

Algorithm 1 BRIDGED CLUSTERING

Input: Input-only set $\mathcal{X} = \{x_1, x_2, \ldots, x_{|\mathcal{X}|}\}$; Output-only set $\mathcal{Y} = \{y_1, y_2, \ldots, y_{|\mathcal{Y}|}\}$; sparse supervised set $\mathcal{S} = \{(x'_i, y'_i)\}_{i=1}^{k}$; Test-set $\mathcal{X}_{\text{test}} = \{\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_{|\mathcal{X}_{\text{test}}|}\}$

Output: $\hat{\mathcal{Y}} = \{\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_{|\mathcal{X}_{\text{test}}|}\}$

1: Apply a clustering algorithm to $\mathcal{X}$ (respectively, $\mathcal{Y}$) to obtain cluster assignments $\mathcal{C}_{\mathcal{X}}$ (respectively, $\mathcal{C}_{\mathcal{Y}}$).
2: Using the supervised set $\mathcal{S}$, learn a mapping $\hat{A}_{x\to y}$ between clusters in $\mathcal{X}$ and clusters in $\mathcal{Y}$.
3: **for** each sample x_i in $\mathcal{X}_{\text{test}}$ **do**
4: Assign x_i to a cluster $c_x = \mathcal{C}_{\mathcal{X}}(x_i)$.
5: Find corresponding cluster in $\mathcal{Y}$: $c_y = \hat{A}_{x\to y}(c_x)$.
6: Predict $\hat{y}_i$ as the centroid of cluster c_y in $\mathcal{Y}$.
7: **end for**
8: **return** Predicted outputs $\hat{\mathcal{Y}}$ for test inputs $\mathcal{X}_{\text{test}}$

- An *input pool* $\mathcal{X} = \{x_i\}_{i=1}^{n_X} \subseteq \mathbb{R}^d$ of n_X i.i.d. draws from the marginal $P^X = \sum_t \pi_t P_t^X$;
- an *output pool* $\mathcal{Y} = \{y_i\}_{i=1}^{n_Y} \subseteq \mathbb{R}^{d'}$ of n_Y i.i.d. draws from the marginal $P^Y = \sum_t \pi_t P_t^Y$;
- a *paired pool* $\mathcal{S} = \{(x'_j, y'_j)\}_{j=1}^{k}$, with $k \ll n_X, n_Y$, drawn i.i.d. from joint $P = \sum_t \pi_t (P_t^X \times P_t^Y)$.

The goal is to predict an output Y from a given input X; the reverse direction (predicting X from Y) is symmetric and follows by swapping roles.

4 BRIDGED CLUSTERING ALGORITHM

BRIDGED CLUSTERING (Algorithm 1) first independently clusters the input space $\mathcal{X}$ and the output space $\mathcal{Y}$. Next, it learns a bridge mapping $\hat{A}$ from the paired samples, where each input cluster is matched to an output cluster via majority vote. Finally, for inference, a new input x is assigned to its nearest input cluster, and the prediction $\hat{y}$ is given by the centroid of the linked output cluster.

Cluster assignments We first run a k-clustering algorithm to partition the dataset into k groups separately on $\mathcal{X}$ and $\mathcal{Y}$ using C centers:

$$\hat{c}_X : \mathbb{R}^d \to [C], \hat{c}_Y : \mathbb{R}^{d'} \to [C].$$

Because cluster labels are only defined up to a permutation of indices, we evaluate their quality via the permutation-invariant mis-clustering rates:

$$\varepsilon_X := \min_{\sigma_X : [C] \to [C]} \Pr\big[\sigma_X(\hat{c}_X(X)) \neq T\big]$$

$$\varepsilon_Y := \min_{\sigma_Y : [C] \to [C]} \Pr\big[\sigma_Y(\hat{c}_Y(Y)) \neq T\big].$$

Let σ_X, σ_Y denote permutations that attain these minima, and let $\sigma_X^{-1}, \sigma_Y^{-1}$ denote their inverses.

Bridging from few pairs On the k paired points, we *match* input-space clusters to output-space clusters by majority vote. For every input cluster $a \in [C]$, mapping

$$\hat{A}(a) \;:=\; \arg\max_{b \in [C]} \sum_{(x', y') \in \mathcal{S}} \mathbf{1}\{\hat{c}_X(x') = a,\ \hat{c}_Y(y') = b\}.$$

For a random latent label T, we call

$$\varepsilon_B := \Pr\Big[\sigma_Y\Big(\hat{A}(\sigma_X^{-1}(T))\Big) \neq T\Big].$$

the *mis-bridging rate*. Algorithm 1 predicts a fresh output by returning the empirical centroid of the output cluster indicated by $\hat{A} \circ \hat{c}_X$. Let $\widehat{\mu}_b^Y$ be the empirical mean of $\mathcal{Y}$-points assigned to cluster b, so our final predictor is

$$\hat{Y}(x) := \widehat{\mu}^{\,Y}_{\hat{A}(\hat{c}_X(x))}.$$

Bidirectionality So far Algorithm 1 predicts output features for test inputs. Using the same cluster-mapping, the algorithm can naturally be extended to predict the input features of output-only datapoints as well (see Appendix C).

5 Algorithmic Variants

To preserve our conceptual simplicity, the main Algorithm 1 adopts general primitives: any clustering method for unsupervised learning and any mapping mechanism for sparse bridging.

Our simple algorithm is meant to be the inspiration and blueprint for a family of output-aware SSL algorithms. To illustrate how Bridged Clustering could be further developed, and to explore the usefulness of these extensions, we articulate and experiment with two algorithmic variants: Soft Bridging, and Clusterwise Regressed Bridging.

5.1 Soft Bridging for Flexible Mapping

When applying our algorithm to datasets without clearly separated categories, clustering primitives may become prone to overlapping or miscounted clusters. Rigid 1-to-1 mapping, as in § 4, becomes vulnerable to misattributed clusters under this setting.

Hence, we design a soft-bridging function as our new mapping primitive: using a small supervised set to build input-output cluster bridges, we construct probabilistic bridges between every possible input-output cluster pair, dynamically weighting each edge by the number of supervised points connecting the clusters. Formally, let $w_{i\to j}$ denote the fraction of supervised points whose inputs fall in input cluster i and whose labels lie in output cluster j, with $\sum_j w_{i\to j} = 1$. For an input x assigned to cluster i, soft bridging predicts a weighted combination of all output cluster centroids:

$$\hat{y}(x) \;=\; \sum_{j=1}^{C} w_{i\to j}\, \bar{y}_j.$$

This method reduces the impact of enforcing a rigid one-to-one bridge when the clustering is imperfect. In practice, the clustering step may return the wrong number of clusters or be unable to distinguish overlapping clusters, which can lead to incorrect cluster correspondences. Soft bridging avoids committing to a single correspondence by distributing probability mass across output clusters, and our experiments show that it is robust on noisy data.

5.2 Clusterwise Regressed Bridging For Refined Prediction

Once a test point x is assigned to an input cluster i and bridged to an output cluster j, 1 returns the output centroid $\widehat{\mu}_j^Y$, collapsing intra-cluster variation. To relax this rigidity, we propose clusterwise regression, deployed as a third step after clustering and bridging. For each cluster pair, we use in-cluster supervised points to fit a linear map $f_{i\to j}(x) \in \mathbb{R}^{d'}$ and predict

$$\hat{y}(x) \;=\; (1-\alpha)\,\widehat{\mu}_j^Y \;+\; \alpha\, f_{i\to j}(x),$$

with $\alpha \in [0,1]$ controlling the strength of the regression refinement. We perform linear regression on the supervised pairs *plus* the C estimated input/output cluster centroids.

Cluster-wise regression provides a natural extension of Bridged Clustering beyond low-supervision and conditionally independent settings. When Y is well approximated as conditionally linear in X given the clustered latent T, i.e., $Y \approx W_t X + b_t$ for $T = t$, sufficient labeled pairs per cluster allow reliable estimation of these cluster-wise maps.

6 Experiments

We test the predictive accuracy of Bridged Clustering and its variants. We use fully paired datasets with known input-output correspondences ($(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$) across various modalities. These datasets simulate independently collected $\mathcal{X}$-only and $\mathcal{Y}$-only repositories by hiding most of the pairings and exposing only a small, randomly selected subset during training. This controlled setup with ground-truth pairs measures performance as a function of the number of labeled pairs.

6.1 DATA

We apply BRIDGED CLUSTERING on four multimodal datasets that connect distinct feature spaces: the *BIOSCAN-5M* dataset that pairs specimen images with DNA barcode sequences of insect samples Gharaee et al. (2024), the *WIT* dataset Srinivasan et al. (2021) pairing images with their text captions on Wikipedia pages, as well as the *Flickr30k* dataset Young et al. (2014) which connects images of people engaged in everyday activities with human-annotated captions and the *COCO* dataset Lin et al. (2014) that connects common object images with text descriptions.

For evaluation, we randomly sample a number of *data groups*, where each group corresponds to a coherent semantic category or label within the dataset (e.g., an insect species in BIOSCAN, or an object category in COCO). Each run includes a subset of these groups (e.g., 5 species of insects from BIOSCAN or 7 categories of objects from COCO) to assess BRIDGED CLUSTERING's ability to recover and align latent structure. We detail the data, processing, and mapping in Appendix A.

6.2 MAIN EXPERIMENT OVERVIEW

We evaluate BRIDGED CLUSTERING's ability to predict output features $\mathcal{Y}$ (e.g., DNA sequences or text captions) from input features $\mathcal{X}$ (e.g., images). Since BRIDGED CLUSTERING is bidirectional, we also run reversed experiments by predicting input features $\mathcal{X}$ from output features $\mathcal{Y}$.

For each dataset, we simulate a disjoint data setting by designating a small subset of datapoints as the supervised set, which contains paired input-output examples. The remaining datapoints are divided into two unpaired sets: one with input-only features and the other with output-only features, mimicking independent data sources. We run experiments in two different settings: transductive, where the model has access to the unlabeled test inputs during training (with test outputs withheld), and inductive, where evaluation is performed on entirely unseen test data. For evaluation, we compute the Mean Squared Error (MSE) between the predicted and the ground-truth output features, as it directly measures prediction accuracy for embeddings (Details in Appendix A).

6.3 BASELINES

We compare BRIDGED CLUSTERING against a comprehensive suite of baselines spanning supervised learning, classical transductive methods, graph-based models, consistency-regularized deep learners, pairwise regression approaches, and unmatched regression techniques (see more information in § 2). Together, they span the full landscape of SSL and related paradigms, whereas all except KNN leverage unlabeled inputs and paired data, and BRIDGED CLUSTERING, KMM, EM, EOT, GW incorporate unlabeled outputs as well. The following list categorizes the baselines we compare BRIDGED CLUSTERING against:

- **Supervised:** K-nearest neighbors (KNN).
- **Transductive SSL:** Laplacian Regularized Least Squares (LapRLS) and Transductive SVR (TSVR).
- **Graph-based:** Graph Convolutional Network (GCN).
- **Consistency-regularized:** FixMatch.
- **Low-supervision regressors:** Twin Neural Network Regression (TNNR), and Uncertainty-Consistent Variational Model Ensembling (UCVME).
- **Unmatched-regression:** Kernel Mean Matching (KMM), Mixture-of-Regressions via Expectation Maximization (EM).
- **Transport:** Entropic OT with Barycentric Mapping (EOT), Gromov-Wasserstein (GW).

6.4 ADDITIONAL EXPERIMENTS FOR ROBUSTNESS

Under real-world data conditions, imperfect cluster qualities may become a greater challenge for our algorithm. For instance, categorical clusters may overlap each other, the number of clusters may grow excessive, and k-clustering primitives may miss-count the number of data groups. Thus, we explicitly design a set of cluster-adversarial experiments.

To target *overlapping* clusters, we design a new experiment by sampling multi-labeled COCO classes that share at least one overlapping label with another sampled class, in contrast to our ear-

(a) Transductive experiments

Experiment	BC	BC (Soft)	BC (Reg)	KNN	FM	LapR.	TSVR	TNNR	UCVME	GCN	KMM	EM	EOT	GW
BIOSCAN	14%	**63%**	14%	0%	1%	0%	0%	0%	6%	0%	2%	0%	0%	0%
BIOSCAN (rev.)	30%	**61%**	0%	0%	0%	0%	0%	0%	3%	0%	4%	1%	0%	0%
COCO	15%	7%	**77%**	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
COCO (rev.)	34%	**55%**	2%	0%	1%	3%	0%	0%	1%	0%	0%	2%	0%	2%
Flickr30k	15%	27%	**40%**	0%	0%	1%	0%	0%	0%	0%	0%	0%	16%	0%
Flickr30k (rev.)	17%	**65%**	0%	0%	1%	1%	0%	0%	2%	0%	0%	2%	0%	13%
WIT	0%	6%	2%	0%	0%	0%	0%	0%	0%	0%	4%	0%	0%	**87%**
WIT (rev.)	0%	6%	2%	0%	6%	0%	0%	0%	3%	6%	2%	0%	0%	**75%**

(b) Inductive experiments

Experiment	BC	BC (Soft)	BC (Reg)	KNN	FM	LapR.	TSVR	TNNR	UCVME	GCN	KMM	EM	EOT	GW
BIOSCAN	13%	**60%**	17%	0%	1%	0%	1%	0%	6%	0%	1%	1%	0%	0%
BIOSCAN (rev.)	28%	**63%**	0%	0%	0%	0%	0%	0%	4%	0%	3%	1%	0%	1%
COCO	18%	12%	**69%**	0%	0%	0%	0%	0%	0%	0%	0%	1%	0%	0%
COCO (rev.)	24%	**56%**	11%	0%	0%	2%	1%	0%	1%	0%	0%	2%	0%	2%
Flickr30k	21%	21%	27%	0%	0%	1%	0%	0%	0%	0%	0%	1%	**28%**	1%
Flickr30k (rev.)	15%	**61%**	1%	0%	1%	0%	2%	0%	1%	0%	0%	2%	0%	16%
WIT	1%	7%	6%	0%	0%	0%	0%	0%	0%	0%	5%	2%	1%	**77%**
WIT (rev.)	0%	8%	4%	0%	4%	0%	0%	0%	4%	3%	4%	0%	0%	**73%**

Table 1: Win-rates across transductive (a) and inductive (b) experiments, rounded to 1%. Bold indicates the best model per dataset (each row is summed to approx. 100%). Win-rates are defined as the model with the lowest MSE across 600 runs in every dataset (30 randomized trials per setting).

lier use of single-labeled categories. Meanwhile, to target *misspecified number* of clusters, we design another experiment in COCO by intentionally mis-specifying the number of clusters in our K-clustering primitives: overcounting by +1/+2 clusters in WIT and undercounting by -1/-2 clusters in Flickr30k. We also stress-test our algorithm's performance by sampling *large number* of data groups for BIOSCAN.

7 Results and Analysis

Each run of our experiments samples a subset of groups (e.g., species in BIOSCAN) from our datasets. We test the robustness of Bridged Clustering by sampling different numbers of data groups $\in \{3, 4, 5, 6, 7\}$ and varying the numbers of supervised samples per cluster $\in \{1, 2, 3, 4\}$

Each setting is run 30 times with different seeds. Table 1 reports win-rates (lowest MSE across 600 runs per dataset), and Figure 2 shows MSE distributions. The complete results of all experiments are found in Appendix Figures 4 to 9. Overall, **Bridged Clustering and variants outperform all baselines across all datasets except WIT**.

7.1 Discussion

Bridged Clustering and its variants outperform all baselines on three of the four datasets. On these three datasets, the improvements are statistically significant; we discuss WIT separately below. In additional experiments for adversity, Bridged Clustering and its variants show less dominant but strong performance (See 14 - 17 in Appendix G).

Robust success Across both directions ($\mathcal{X} \to \mathcal{Y}$, $\mathcal{Y} \to \mathcal{X}$) in BIOSCAN, Flickr30k, and COCO, Bridged Clustering algorithms achieves consistently low MSE, **outperforming all baselines over varied cluster sizes and supervision levels**. They not only surpass all SSL baselines by leveraging additional output-only data $\mathcal{Y}$, but also outperform KMM, EM, EOT, and GW (although these methods use full training data), underscoring the strength of our simpler approach. In WIT, Bridged Clustering remains competitive. Adversarial conditions do bring more challenges to our algorithm, but Bridged Clustering (especially the Soft Bridging) maintains overall success.

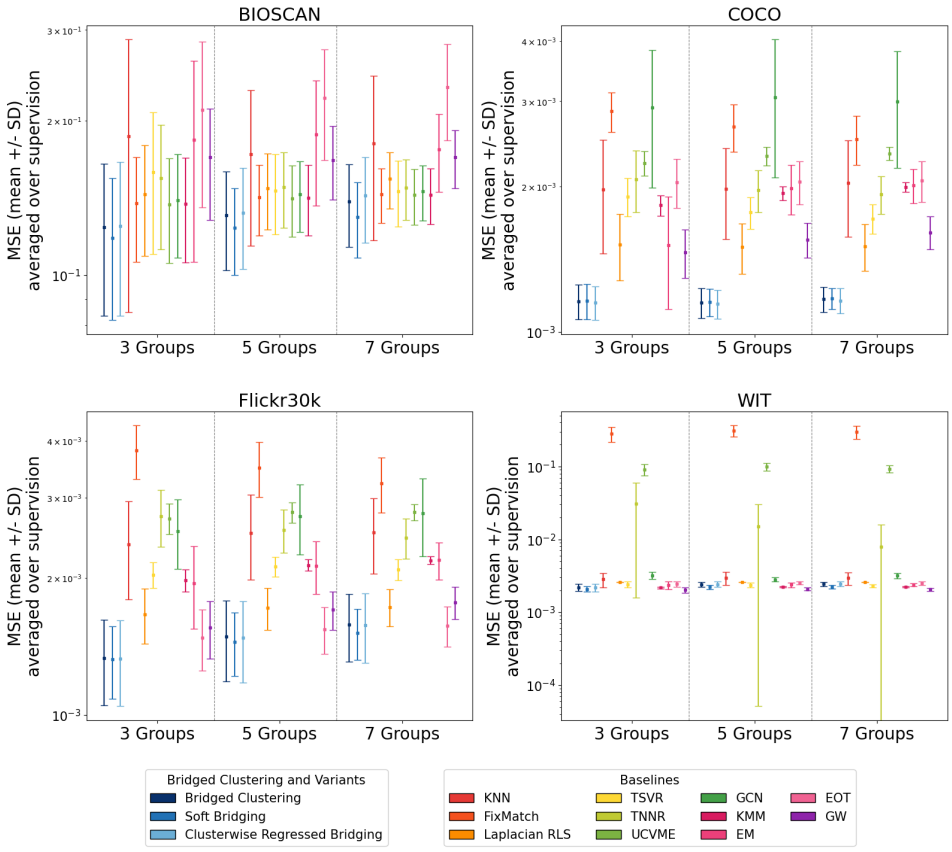


Figure 2: MSE distribution of different models for forward experiments in the inductive setting. The distribution plots aggregates runs over 1, 2, 3, and 4 supervised samples per cluster. See more detailed MSE figures in the Appendix E, including reversed experiments and transductive experiments.

Cluster quality and accuracy BRIDGED CLUSTERING's performance depends on two factors: recovering latent structure via independent clustering (cluster quality) and aligning clusters with limited supervision (bridging accuracy). We observe that **cluster quality is negatively correlated with model loss**, measured using adjusted mutual information (AMI) (Figure 10 in Appendix F). WIT's low mutual information scores suggests that weaker clustering explains BRIDGED CLUSTERING's lower performance on this dataset. We also observe a positive correlation between bridging accuracy and cluster quality in Figure 12, indicating that these two performance-relevant factors are correlated, a connection that'll be rediscussed in § 8.

Transductive Versus Inductive: The performance of Bridged Clustering across both transductive and inductive settings are consistently competitive, with slightly more wins in the transductive tests. Given that our clustering-based method naturally takes advantage of inference points to better learn the cluster geometry in the transductive settings, this marginal win is expected.

Higher risk with more data groups BRIDGED CLUSTERING's relative weakness in high-cluster regimes stems from its reliance on bijection input-output mapping. As cluster count increases, the space of possible mappings grows exponentially, making accurate alignment harder (see Figure 13 in Appendix F). Baseline methods without such explicit mappings prove more robust to increases in cluster count, highlighting BRIDGED CLUSTERING's tradeoff: it **excels in low-supervision, low-cluster settings, but becomes more sensitive as structural complexity increases**.

Variant Robustness In both the main and the adversarial experiments, Soft Bridging Variant often outperform the basic algorithm and most other baselines, suggesting the strength of this algorithmic variant and the adaptivity of BRIDGED CLUSTERING as a template for crafting diverse algorithms

for challenging data. Note that Soft Bridging is a natural and particularly strong solution to overlapping clusters in COCO and overnumbered clusters in BIOSCAN.

8 THEORETICAL DISCUSSION

We provide the mathematical principals behind BRIDGED CLUSTERING, identifying key factors for its effectiveness. We also provide an asymptotic runtime analysis in Appendix D. Representation quality and scarce supervision influence the final accuracy of BRIDGED CLUSTERING through several key sources of error, including the input mis-clustering (ε_X), output mis-clustering (ε_Y), and mis-bridging (ε_B). The main takeaways of this section are that (1) stronger embeddings that enlarge inter-clustering separation directly reduce ε_X and ε_Y, and (2) once clusters are reliable, even a small handful of paired examples suffice to learn the sparse cluster bridge with exponentially small error.

Mis-clustering under sub-Gaussian mixtures. Suppose the input and output distributions are σ^2–sub-Gaussian with means μ_t^X, μ_t^Y. Define the minimum separation

$$\Delta_X = \min_{t \neq s \in [C]} \|\mu_t^X - \mu_s^X\|, \Delta_Y = \min_{t \neq s \in [C]} \|\mu_t^Y - \mu_s^Y\|.$$

It is known that under Lloyd's algorithm, under good initialization, C-means clustering on n_X unlabeled inputs satisfies

$$\varepsilon_X \leq \exp\left(-\frac{\Delta_X^2}{16\sigma^2}\right) \quad \text{for all } s \geq 4 \log n_X$$

with probability at least $1 - \nu - \frac{4}{n_X} - 2\exp(-\Delta_X/\sigma)$, where ν is the initializer failure rate Lu & Zhou (2016). An identical bound holds for ε_Y. Thus, any improvement in embeddings that increases Δ_X or Δ_Y sharpens the bound without changing the algorithm. Consistent with this theory, we observe a strong negative correlation between cluster quality (AMI) and predictive loss (Appendix F).

Efficient bridging from few pairs. Once the mis-clustering rates $\varepsilon_X, \varepsilon_Y$ are controlled, learning the bridge becomes simple. As observed in our experiments, there's a positive correlation between bridging accuracy and cluster quality. For a supervised pair $(X, Y)|T \in \mathcal{S}$, consider the events $\{\sigma_X(\hat{c}_X(X)) = T\}$ and $\{\sigma_Y(\hat{c}_Y(Y)) = T\}$, which correspond to correct cluster assignments in the input and output spaces, controlled via $\varepsilon_X, \varepsilon_Y$. If both events hold with sufficiently high probability ($> 50\%$), then a majority vote over just a few labeled pairs will suffice to recover the correct cluster-to-cluster assignment, resulting in a low mis-bridging error ε_B. Empirically, we find that even one or two examples per cluster are enough for reliable bridging, explaining why BRIDGED CLUSTERING remains effective with less than 1% paired supervision (§ 6).

9 CONCLUSION & FUTURE WORK

BRIDGED CLUSTERING offers a simple new perspective for SSL by elevating *output-only* data from a peripheral role to a central asset. By decoupling input and output representations and connecting them via a sparse, interpretable bridge, **our framework addresses a gap between classic input-centric SSL and modern data ecosystems with abundant but misaligned observations**.

Empirically, BRIDGED CLUSTERING performs competitively across diverse domains and modalities. The method is model-agnostic, computationally efficient, and supports bidirectional inference without retraining. By separating representation learning from supervision, it uncovers latent structure in both input and output spaces and aligns them with minimal labels. **Concept-level alignment between independently clustered spaces provides a strong inductive bias, enabling generalization without dense supervision**, which is especially valuable in settings with limited labeled data, high domain variability, or distribution shifts.

Our framework is naturally extensible while preserving interpretability and efficiency. E.g., algorithmic variants such as Soft-Bridging and Cluster-wise Regression bring robustness in adversarial data conditions. Future work can explore combination of BRIDGED CLUSTERING with unsupervised techniques like manifold learning or self-supervised embeddings, in effort to build more flexible semi-supervised learning systems capable of leveraging richly available but misaligned data.

AI USE STATEMENT

We used ChatGPT to aid and polish this paper, including grammatical and writing quality checks. We also used Codex to help with writing code.

ETHICS STATEMENT

BRIDGED CLUSTERING primarily advances methodology in machine learning and does not directly involve human subjects or sensitive personal data. Nonetheless, we acknowledge that methods for automated clustering and prediction can be applied in real-world settings with potential social impact.

The design of BRIDGED CLUSTERING emphasizes interpretability and low supervision, which can help mitigate risks associated with opaque or biased models. By explicitly linking clusters and maintaining cluster-level transparency, users can inspect and understand the learned correspondences, reducing the likelihood of unintended harm from unexamined correlations.

We caution that, like any predictive system, applying BRIDGED CLUSTERING to high-stakes domains (e.g., hiring, law enforcement, healthcare) requires careful consideration of fairness, privacy, and potential biases in the data. Our experiments are conducted on publicly available benchmark datasets, and no sensitive or personally identifiable information is used.

REPRODUCIBILITY STATEMENT

All our experiments are fully reproducible. Reproducibility information about our computational experiments is available at our code repository, including method implementations, data processing, final hyperparameters, statistical tests and more.

REFERENCES

Wafa'Za'al Alma'aitah, Addy Quraan, Fatima N AL-Aswadi, Rami S Alkhawaldeh, Moutaz Alazab, and Albara Awajan. Integration approaches for heterogeneous big data: A survey. *Cybernetics and Information Technologies*, 24(1), 2024.

Pablo Millan Arias, Niousha Sadjadi, Monireh Safari, ZeMing Gong, Austin T. Wang, Joakim Bruslund Haurum, Iuliia Zarubiieva, Dirk Steinke, Lila Kari, Angel X. Chang, Scott C. Lowe, and Graham W. Taylor. BarcodeBERT: Transformers for biodiversity analysis. *Bioinformatics Advances*, 6(1), 2026.

Mona Azadkia and Fadoua Balabdaoui. Linear regression with unmatched data: A deconvolution perspective. *Journal of Machine Learning Research*, 25(197):1–55, 2024.

Maria-Florina Balcan and Avrim Blum. A discriminative model for semi-supervised learning. *Journal of the ACM (JACM)*, 57(3):1–46, 2010.

Maria-Florina Balcan, Avrim Blum, and Ke Yang. Co-training and expansion: Towards bridging theory and practice. In *Advances in Neural Information Processing Systems*, 2004.

Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of machine learning research*, 7, 2006.

David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. *Advances in neural information processing systems*, 32, 2019.

Steffen Bickel and Tobias Scheffer. Multi-view clustering. In *Proceedings of the IEEE International Conference on Data Mining*, 2004.

Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pp. 92–100, 1998.

Alexandra Carpentier and Teresa Schlüter. Learning relationships between data obtained independently. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, 2016.

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

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013.

Weihang Dai, Xiaomeng Li, and Kwang-Ting Cheng. Semi-supervised deep regression with uncertainty consistency and variational model ensembling via bayesian neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 7304–7313, 2023.

Zhiyuan Dang, Cheng Deng, Xu Yang, Kun Wei, and Heng Huang. Nearest neighbor matching for deep clustering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 13693–13702, 2021.

Sanjoy Dasgupta, Michael Littman, and David McAllester. Pac generalization bounds for co-training. *Advances in neural information processing systems*, 14, 2001.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society: series B (methodological)*, 39(1), 1977.

Zahra Gharaee, Scott C Lowe, ZeMing Gong, Pablo M Arias, Nicholas Pellegrino, Austin T Wang, Joakim B Haurum, Iuliia Zarubiieva, Lila Kari, Dirk Steinke, et al. Bioscan-5m: A multimodal dataset for insect biodiversity. *Advances in Neural Information Processing Systems*, 37:36285–36313, 2024.

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

Pin-Yen Huang, Szu-Wei Fu, and Yu Tsao. Rankup: Boosting semi-supervised regression with an auxiliary ranking classifier. *Advances in Neural Information Processing Systems*, 37:107444–107468, 2024.

Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 172–189, 2018.

Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242*, 2016.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. Unsupervised machine translation using monolingual corpora only. *arXiv preprint arXiv:1711.00043*, 2017.

Eric Lei, Kyle Miller, Michael R. Pinsky, and Artur Dubrawski. Characterization of hemodynamic signal by learning multi-view relationships. *arXiv preprint arXiv:1709.05602*, 2017.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

Carolus Linnaeus. *Systema Naturae per regna tria naturae, secundum classes, ordines, genera, species; cum characteribus, differentiis, synonymis, locis*, volume 1. apud JB Delamolliere, 1789.

Yu Lu and Harrison H Zhou. Statistical and computational guarantees of lloyd's algorithm and its variants. *arXiv preprint arXiv:1612.02099*, 2016.

Roger Mark. The story of mimic. In *Secondary Analysis of Electronic Health Records*, pp. 43–49. Springer, 2016.

Facundo Mémoli. Gromov-wasserstein distances and the metric approach to object matching. *Foundations of computational mathematics*, 11:417–487, 2011.

Daniel G Mulcahy. Specimen identifiers: Linking tissues, dna samples, and sequence data to voucher specimens in publicly accessible databases. *Biodiversity Information Science and Standards*, 6, 2022.

Gabriel Peyré and Marco Cuturi. Computational optimal transport: With applications to data science. *Foundations and Trends in Machine Learning*, 11(5-6):355–607, 2019.

Gabriel Peyré, Marco Cuturi, and Justin Solomon. Gromov-wasserstein averaging of kernel and distance matrices. In *International conference on machine learning*, pp. 2664–2672. PMLR, 2016.

I Made Putrama and Péter Martinek. Heterogeneous data integration: Challenges and opportunities. *Data in Brief*, 56, 2024.

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

Sujeevan Ratnasingham, Catherine Wei, Dean Chan, Jireh Agda, Josh Agda, Liliana Ballesteros-Mejia, Hamza Ait Boutou, Zak Mohammad El Bastami, Eddie Ma, Ramya Manjunath, et al. Bold v4: A centralized bioinformatics platform for dna-based biodiversity data. In *DNA barcoding: Methods and protocols*, pp. 403–441. Springer, 2024.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pp. 3982–3992, 2019.

Melanie Rey and Volker Roth. Copula mixture model for dependency-seeking clustering. *arXiv preprint arXiv:1206.6433*, 2012.

Philippe Rigollet and Jonathan Weed. Uncoupled isotonic regression via minimum wasserstein deconvolution. *Information and Inference: A Journal of the IMA*, 8(4):691–717, 2019. doi: 10.1093/imaiai/iaz006.

Eric W Sayers, Mark Cavanaugh, Karen Clark, James Ostell, Kim D Pruitt, and Ilene Karsch-Mizrachi. Genbank. *Nucleic acids research*, 47:D94–D99, 2019.

Erich Schubert, Jörg Sander, Martin Ester, Hans Peter Kriegel, and Xiaowei Xu. Dbscan revisited, revisited: why and how you should (still) use dbscan. *ACM Transactions on Database Systems*, 42(3):1–21, 2017.

Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems*, 33:596–608, 2020.

Krishna Srinivasan, Karthik Raman, Jiecao Chen, Michael Bendersky, and Marc Najork. Wit: Wikipedia-based image text dataset for multimodal multilingual machine learning. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, pp. 2443–2449, 2021.

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

Jesper E Van Engelen and Holger H Hoos. A survey on semi-supervised learning. *Machine learning*, 109:373–440, 2020.

Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. Scan: Learning to classify images without labels. In *European conference on computer vision*, pp. 268–285. Springer, 2020.

Jaakko Viinikanoja, Arto Klami, and Samuel Kaski. Variational bayesian mixture of robust cca models. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 370–385. Springer, 2010.

Sebastian J Wetzel, Roger G Melko, and Isaac Tamblyn. Twin neural network regression is a semi-supervised regression algorithm. *Machine Learning: Science and Technology*, 3(4):045007, 2022.

Xiangli Yang, Zixing Song, Irwin King, and Zenglin Xu. A survey on deep semi-supervised learning. *IEEE transactions on knowledge and data engineering*, 35(9):8934–8954, 2023.

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the association for computational linguistics*, 2:67–78, 2014.

Dengyong Zhou, Olivier Bousquet, Thomas Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. *Advances in neural information processing systems*, 16, 2003.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

Xiaojin Zhu and Zoubin Ghahramani. Learning from labeled and unlabeled data with label propagation, 2002.

A Experiment Details

This appendix details the embedding models, clustering algorithms, and parameters used.[1]

A.1 BIOSCAN-5M

BIOSCAN-5M contains the specimens images, taxonomic labels, and raw nucleotide barcode sequences of over 5 million insect samples Gharaee et al. (2024). For experimentation, our models will be given the encoded specimens images (x) to predict the encoded nucleotide barcodes (y), based on the latent taxonomic labels (t). Every sample is labeled according to the hierarchical Linnaean taxonomy: class, order, family, genus, and species. Linnaeus (1789).

We set up our dataset based on the test split of the original BIOSCAN-5M dataset, which is representative of the entire dataset and sizable (39,373) for our purposes. According to the lowest-level taxonomy label of "species", we can divide the dataset into hundreds of homogeneous species groups. Among those species, we randomly choose C groups for each experiment run, prioritizing sampling species with different "family" labels, and then randomly draw 200 samples from each of the selected species groups. By selecting our species groups in this way, we approximately meet our theoretical assumption of sub-Gassian mixtures.

Within each group, samples are randomly assigned according to some fixed ratio to one of the three data-splits: input-only, output-only, and supervised. For consistency, we set the output-only split to be 10%, and vary the supervised percentage, using the rest as input-only data (When testing the inverse direction, we keep the input-only split to be 10%, vary the supervised percentage, and use the rest as output-only data). Pooling all data, we remove the species information from the samples, hoping that our clustering procedures can group our data roughly according to latent species affiliations.

To convert the raw data into cluster-able data that we can easily train and evaluate on, we encode the specimens images with a Resnet50 image encoder He et al. (2016) and the DNA barcodes with BarcodeBERT Arias et al. (2026). We also narrow our sampling to the Insecta Class since it contains 98% of the BIOSCAN-5M data.

[1]The runtime and memory requirements depend heavily on the chosen clustering algorithm; our current implementation is not optimized, which complicates empirical evaluation.

A.2 WIT

Wikipedia Image-Text (WIT) Srinivasan et al. (2021) Dataset is a large curated set of more than 37 million image-text associations extracted from Wikipedia articles. Each entry of the dataset contains an Wikipedia image with its image caption, as well as information of all pages containing this particular image.

To curate our dataset, we select the first 100,000 samples from WIT, and encode the text captions with Sentence-BERTReimers & Gurevych (2019).

We further extract the English page titles associated with every image and encode them with Sentence-BERT. We use DBSCANSchubert et al. (2017) to discover dense clusters of datapoints with neighboring page title encodings, which suggests that these datapoints are associated with a latent category of Wikipedia pages. To further ensure closeness, we add another eligibility requirement for the clusters: every member in the cluster must share one word in their page title. Among these dense clusters retrieved, we randomly sample C clusters, choose 25 samples from each cluster to be included in each run, and split them into three data-splits as in BIOSCAN (input-only split containing only images, output-only split containing only image captions, and the fully supervised split). This time we adjust the output-only percentage to 20%, as the cluster sizes become small and it'll be challenging to form effective output clusters with only 10% data. Pooling all data, we remove the page title information from the samples, hoping that our clustering procedures can group our data roughly according to the latent Wikipedia page categories.

A.3 FLICKR30K

The Flickr30k dataset Young et al. (2014) is comprised of 31,783 images that capture people engaged in everyday activities and events. For each image, the dataset provides 5 human-annotated descriptive captions.

We select the first 10,000 samples from Flickr30k, and split each sample into 5 datapoints, so that each entry contains one unique caption and one (possibly repeated) image.

In close parallel to the WIT experiment, we encode the images with ResNet50 and the text captions with Sentence-BERT. To select meaningful families of data, we further encode the text captions with TF-IDF and perform clustering, so that caption that share key words would be clustered close together. We then use DBSCAN to discover dense clusters of datapoints, randomly sample C clusters and sample 25 datapoints per cluster to be included each experiment run. We split the data in each cluster into three data-splits as before, then pool all data for the experiment.

A slight difference from WIT is in caption prediction, where we measure the distance between our model-predicted caption and the closest of the 5 captions that image corresponds to.

A.4 COCO

The Microsoft Common Objects in Context Dataset (COCO Lin et al. (2014)) is another image-text dataset. Every image in the dataset is paired with an array of text annotations, as well as a list of categorical labels, corresponding to different objects present in the image.

Our curated dataset is extracted from the validation split of the original COCO dataset. We sample all datapoints with only one categorical label in their list, and group them according to their categorical labels. In each experiment run, we randomly sample C groups and sample 40 datapoints per group, then follow the experimental procedure as exactly described in Flickr30k. Since clusters sizes are bigger again, we adjust the output-only percentage back to 10%.

A.5 BASELINES

We adapted the baseline methods to our experiment environment, keeping faithful to the original methods and ensuring conceptual correctness. We allocated a reasonable fine-tuning budget for our baseline methods with a small grid across key hyper-parameters. Since BRIDGED CLUSTERING does not require any finetuning, we didn't spend excessive resource in the fine-tuning of the baselines, so the baselines used in our experiments may not be the optimized version. For baseline

Method	Grid
FixMatch	`lr={1e-4, 3e-4, 1e-3}; batch_size={32, 64}; alpha_ema={0.99, 0.999}; lambda_u_max={0.5, 1.0}; rampup_length={10, 30}; conf_threshold={0.05, 0.1}`
LapRLS	`lam={1e-5, 1e-2, 3e-2, 1e-1, 3e-1, 1}; gamma={1e-3, 1e-2, 1e-1, 1}; k={5, 10, 20}; sigma={None, 2.0, 5.0}`
TNNR	`rep_dim={32, 64, 128}; beta={0.01, 0.1, 1.0}; lr={1e-4, 3e-4, 1e-3}; targeted WIT refresh: rep_dim={16, 256}; beta={0.0, 0.3}; lr={3e-3}; epochs={12, 25}; n_loops={1, 2}; max_pairs={512, 1024}; input_components={64, 128}; target_components={32, 64}`
TSVR	`C={0.1, 1, 10}; epsilon={0.01, 0.1}; lambda_w={1e-4, 1e-3, 1e-2}; lambda_graph={0, 0.1, 1, 10}; n_neighbors={5, 10}; fixed: lr=1e-2, steps=500, include_test_inputs=True`
UCVME	`lr={1e-4, 3e-4, 1e-3}; w_unl={1, 5, 10}; mc_T={5, 10}`
GCN	`hidden={32, 64, 128}; dropout={0.0, 0.1, 0.3}; lr={1e-3, 3e-3}; targeted COCO/Flickr refresh: hidden={256}; dropout={0.05, 0.2}; lr={1e-2}; k={5, 10, 20}; weight_decay={0.0, 5e-4}; input_components={32, 64, 128}; target_components={16, 32, 64}; epochs={12, 25}`
KMM	`alpha={1e-2, 1e-1}; kmm_B={100, 1000}; kmm_eps={1e-3, 1e-2}; sigma={0.5, 1.0}`
EM	`n_components={2, 3}; max_iter={100, 200}; tol={1e-3, 1e-4}; eps={1e-3, 1e-4}`
EOT	`eps={1e-3, 1e-2, 1e-1, 1, 10}; ridge_alpha={1e-2, 1e-3, 1e-4}; tol={1e-5, 1e-7, 1e-9}`
GW	`max_iter={200, 400, 800}; tol={1e-5, 1e-7, 1e-9}`

Table 2: Hyperparameter grids used for baseline fine-tuning. Each baseline has a small grid over key hyperparameters; BRIDGED CLUSTERING and KNN require no tuning. Iterative methods use a generous cap, while BRIDGED CLUSTERING typically converges within ≤ 100 iterations. Hyperparameters were selected on small randomly-seeded validation splits per dataset and direction, with lowest MSE as criterion.

methods that involve iterations, we set a reasonable and generous maximum iteration according to the runtime of each algorithm, as compared to 100 or less maximum iterations for BRIDGED CLUSTERING. Please see our code for additional details. We also attach the tuning grid for reference.

A.6 EVALUATION

We compute MSE in the output embedding space between predicted and ground-truth embeddings. For single-target experiments (forward BIOSCAN and WIT, and all reverse experiments), we compare each predicted output embedding directly to the corresponding ground-truth output embedding. For multi-caption experiments (forward COCO/Flickr), each test item retains its set of caption embeddings; for each prediction, we choose the ground-truth caption embedding closest to that prediction and compute MSE between the prediction and that selected caption embedding.

B ON THE IMPORTANCE OF REPRESENTATION OR BRIDGED CLUSTERING

Following our algorithm, we independently cluster the input space (e.g., encoded images) and the output space (e.g., encoded DNA or text). In expectation, effective clustering shall retrieve latent class affiliations. For example, in the BIOSCAN experiment where the latent variable is "species", samples assigned to the same input cluster shall have the same species label, as with output cluster. We measure this clustering effectiveness using Adjusted Mutual Information (AMI) between cluster assignments and latent labels for both inputs and outputs.

Since our algorithm is model-agnostic, we are free to experiment with a wide range of clustering methods. As we know the number of latent variables included in every experiment, hence the ideal

number of clusters, k-clustering methods are natural choices: k-means, spectral clustering, and gaussian mixture model. We also explored agglomerative clustering as a baseline. As we discovered over iterative testing, constrained k-means with balanced cluster sizes is our best performer as measured through AMI, possibly due to the fact that we curated our dataset with equal species-cluster sizes.

Another metric that's directly related to the algorithm's effectiveness is the accuracy of bridged association learning. After clustering, as each supervised point is associated with one input and one output cluster, we obtain a voted mapping between input and output clusters. We measure the accuracy of this mapping by comparing it to the ground-truth. Since every input/output cluster roughly corresponds to one distinct latent variable, with high probability every latent variable will dominate exactly one input cluster and one output cluster, allowing us to obtain a non-overlapping ground-truth mapping through latent-by-latent reconstruction. After trials with different voting mechanisms, including Margin-based voting and Hungarian Algorithm, the most accurate mechanism turns out to be simple majority-based voting.

C Bridged Clustering Algorithm (Bidirectional)

Algorithm 1 can be extended in practice to predict in the inverse direction, using the bidirectional property of Bridged Clustering.

Algorithm 2 Bridged Clustering Algorithm (Bidirectional)

Input: Input-only set $\mathcal{X} = \{x_1, x_2, \ldots, x_{|\mathcal{X}|}\}$; Output-only set $\mathcal{Y} = \{y_1, y_2, \ldots, y_{|\mathcal{Y}|}\}$; sparse supervised set $\mathcal{S} = \{(x'_i, y'_i)\}_{i=1}^{k}$; Input Test-set $\mathcal{X}_{\text{test}} = \{x_1, x_2, \ldots, x_{|\mathcal{X}_{\text{test}}|}\}$; Output Test-set $\mathcal{Y}_{\text{test}} = \{y_1, y_2, \ldots, y_{|\mathcal{Y}_{\text{test}}|}\}$ (Note: $\mathcal{X}_{\text{test}} = \mathcal{X}$, $\mathcal{Y}_{\text{test}} = \mathcal{Y}$ in the *transductive* setting)

Output: $\hat{\mathcal{Y}} = \{\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_{|\mathcal{X}_{\text{test}}|}\}$, $\hat{\mathcal{X}} = \{\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_{|\mathcal{Y}_{\text{test}}|}\}$

1: **Call** Algorithm 1
2: Using the supervised set $\mathcal{S}$, learn the reversed mapping $\hat{A}_{y \to x}$ between clusters in $\mathcal{Y}$ and clusters in $\mathcal{X}$.
3: **for** each sample y_i in $\mathcal{Y}_{\text{test}}$ **do**
4: Assign y_i to a cluster $c_y = \mathcal{C}_{\mathcal{Y}_{\text{test}}}(y_i)$.
5: Find corresponding cluster in $\mathcal{X}$: $c_x = \hat{A}_{y \to x}(c_y)$.
6: Predict $\hat{x}_i$ as the centroid of cluster c_x in $\mathcal{X}$.
7: **end for**
8: **return** Predicted outputs $\hat{\mathcal{Y}}$ for given inputs, Predicted inputs $\hat{\mathcal{X}}$ for given outputs.

D Runtime and Memory Complexity

We break down the cost of Bridged Clustering into clustering, bridge learning, and inference. To remain solver-agnostic, let $f(n, C, d)$ denote the time our chosen algorithm takes to cluster n points in $\mathbb{R}^d$ into C clusters. For Bridged Clustering:

$$\text{training} = f(n_X, C, d) + f(n_Y, C, d') + O(k),$$

$$\text{per-query inference} = O(Cd + d'),$$

with memory $O(n_X d + n_Y d' + C(d+d') + C^2)$.[2] Instantiating f: Lloyd/k-means++ gives $O(nCdI)$ where I is the number of iterations; mini-batch k-means $O(mCdI)$ with batch $m \ll n$; pruned k-means $O(nC'dI)$ with $C' \leq C$ effective distance calls. Thus, training is linear in n_X, n_Y when $C \ll n$, and inference requires only $O(C)$ distance computations instead of dense cross-space objects.

Now, as we found in our experiments in § 6, the transport-based methods, EOT and GW, are the closest competitors to our method. We compare our method asymptotically to the standard implementations of EOT and GW. This analysis is conducted in the transductive setting, though the asymptotic conclusions extend similarly to the inductive case. Let S denote the number of Sinkhorn iterations in EOT, and T_{GW} the number of outer GW iterations; we report only the dominant terms.

[2] If only a mapping $[C] \to [C]$ is stored, the bridge term is $O(C)$; $O(C^2)$ is a safe upper bound for storing a sparse $C \times C$ vote/mapping matrix.

Method	Training time	Memory
BC (ours)	$f(n_X, C, d)+f(n_Y, C, d')+O(k)$	$O(n_X d + n_Y d' + C(d+d')+C^2)$
EOT	$O(S\, n_X n_Y)$	$O(n_X n_Y)$
GW	$O(T_{\text{GW}}(n_X^2 n_Y + n_X n_Y^2))$	$O(n_X^2 + n_Y^2 + n_X n_Y)$

Since both transport-based methods involve Sinkhorn iterations and dense couplings, they incur at least quadratic cost in both time and memory, which scales quickly with data increase. In contrast, BRIDGED CLUSTERING with a fast clustering routine (e.g., mini-batch k-means) achieves training time linear in n_X, n_Y and memory linear in the raw data size plus a (sparse) $C\times C$ bridge, making it more efficient in both runtime and memory than transport-based baselines in large-pool regimes.

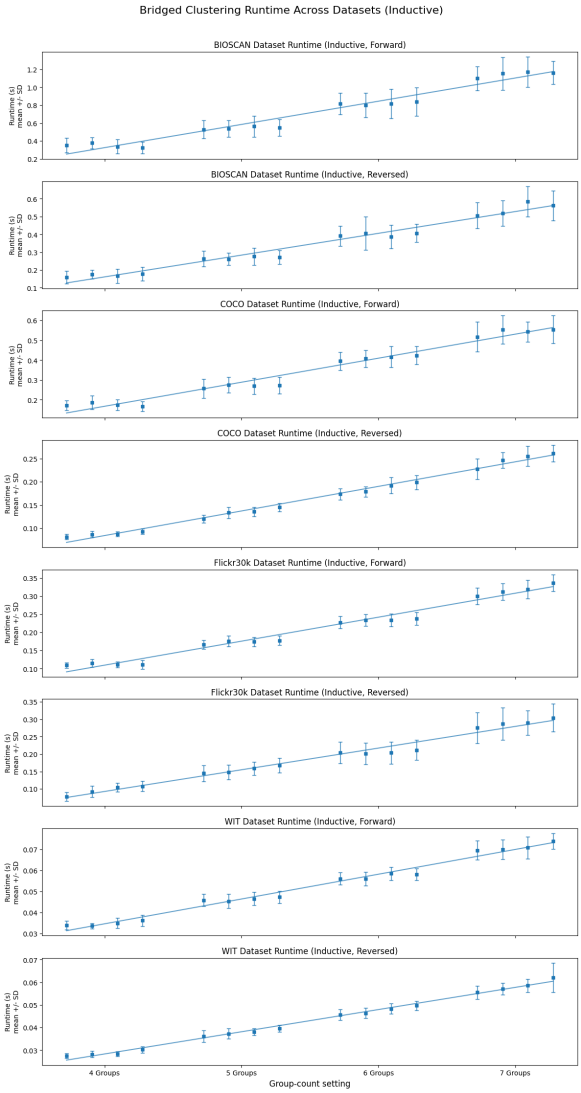


Figure 3: Empirical Runtime shows linear growth, mirroring theoretical bound (Inductive Experiments shown here. Transductive shows similar trend).

E DETAILED EXPERIMENT RESULTS: MSE

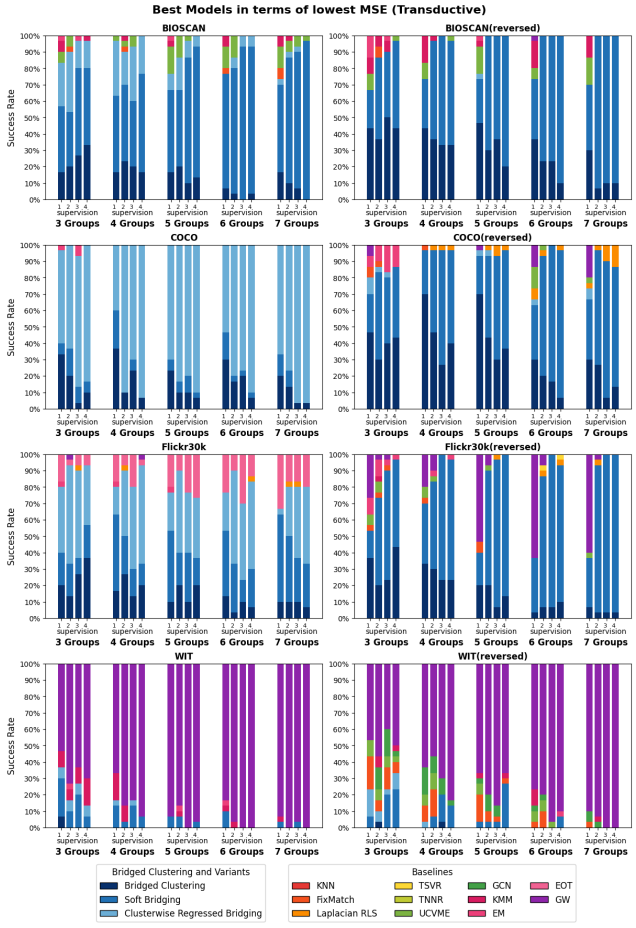


Figure 4: Tansductive Experiment: Best models in terms of lowest MSE, computed across 30 randomized trials per setting. Each bar represents the 30 trials of one setting. For example, if BRIDGED CLUSTERING achieves the lowest MSE among all models in 15 out of the 30 trials for some setting, the bar that corresponds to that setting will be colored 50% blue. The 1,2,3,4 ticks on the bottom represent the settings with 1,2,3,4 supervised samples per cluster.

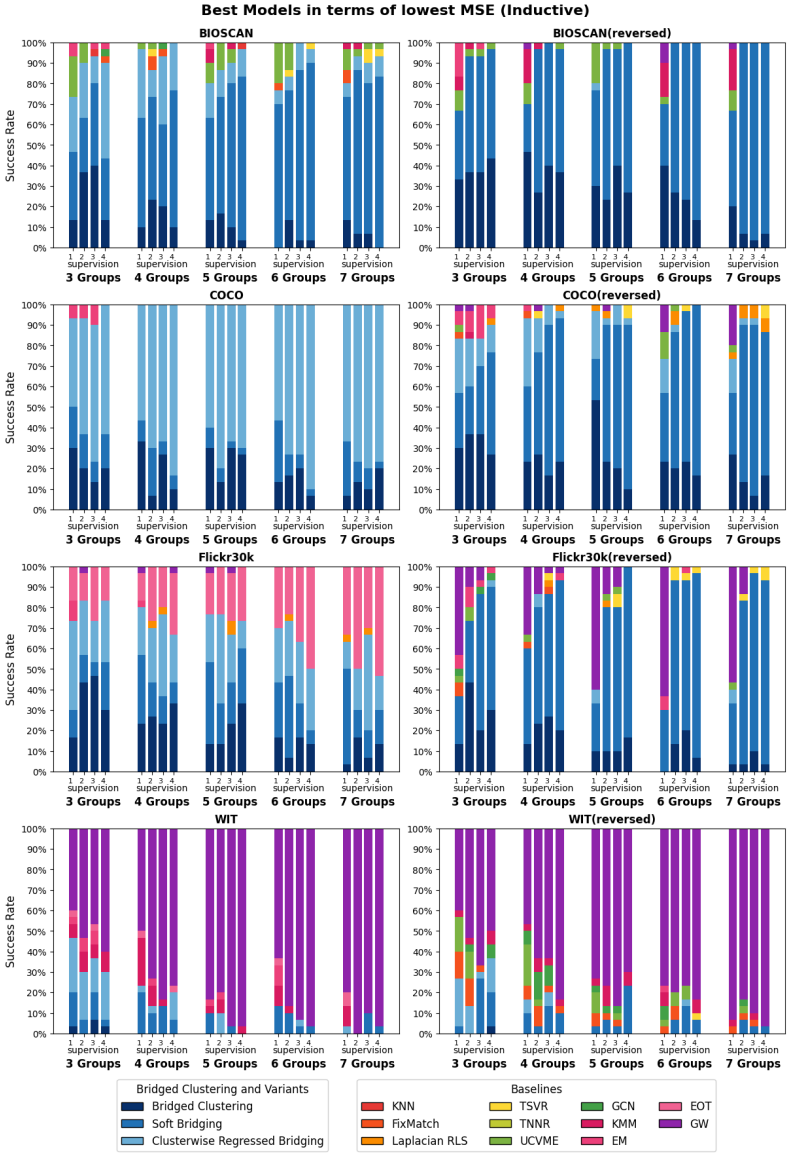


Figure 5: Inductive Experiment: Best models in terms of lowest MSE, computed across 30 randomized trials per setting. Each bar represents the 30 trials of one setting. For example, if BRIDGED CLUSTERING achieves the lowest MSE among all models in 15 out of the 30 trials for some setting, the bar that corresponds to that setting will be colored 50% blue. The 1,2,3,4 ticks on the bottom represent the settings with 1,2,3,4 supervised samples per cluster.

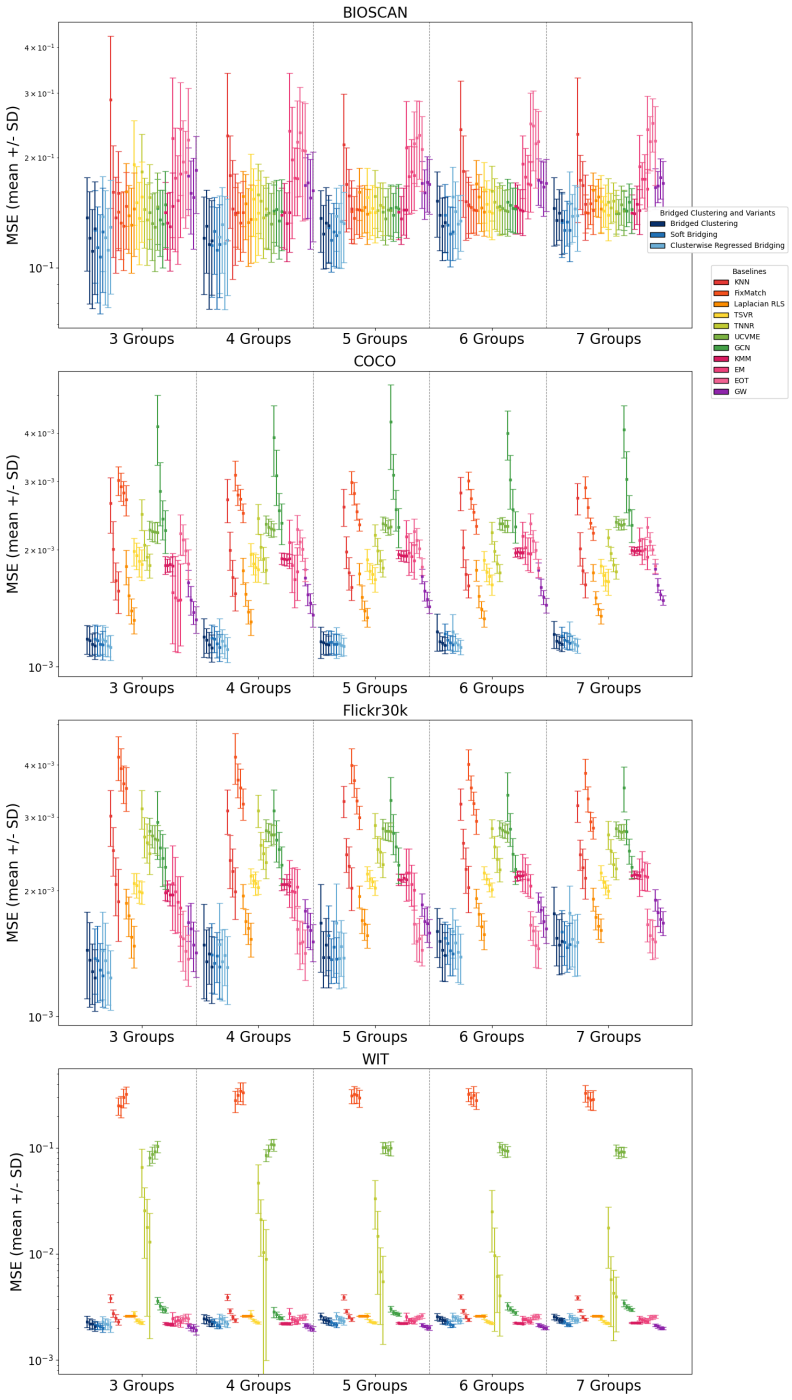


Figure 6: MSE distribution of different models in the transductive setting. The 4 distribution plots of the same color represent the settings with 1,2,3,4 supervised samples per cluster.

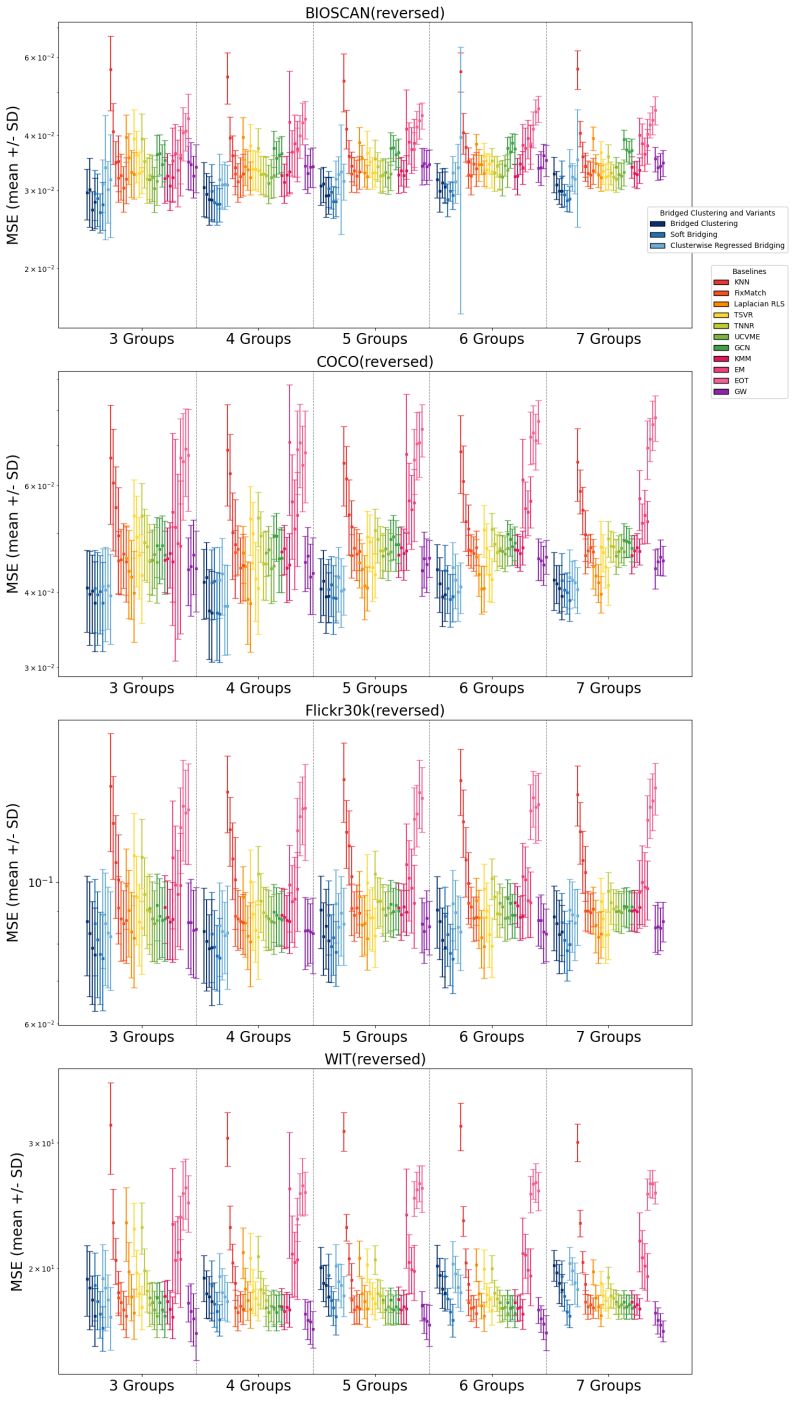


Figure 7: MSE distribution of different models in reversed experiments in the transductive setting. The 4 distribution plots of the same color represent the settings with 1,2,3,4 supervised samples per cluster.

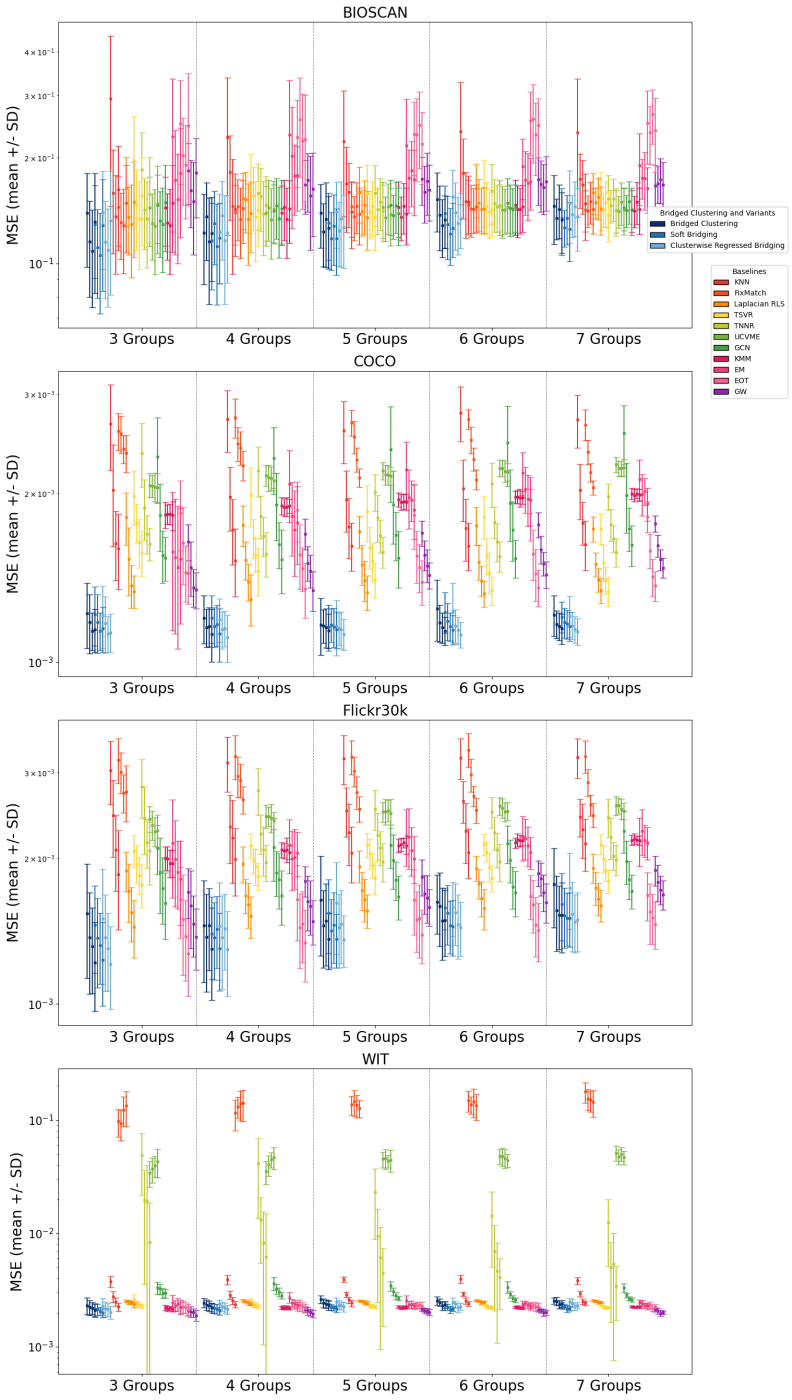


Figure 8: MSE distribution of different models in the inductive setting. The 4 distribution plots of the same color represent the settings with 1,2,3,4 supervised samples per cluster.

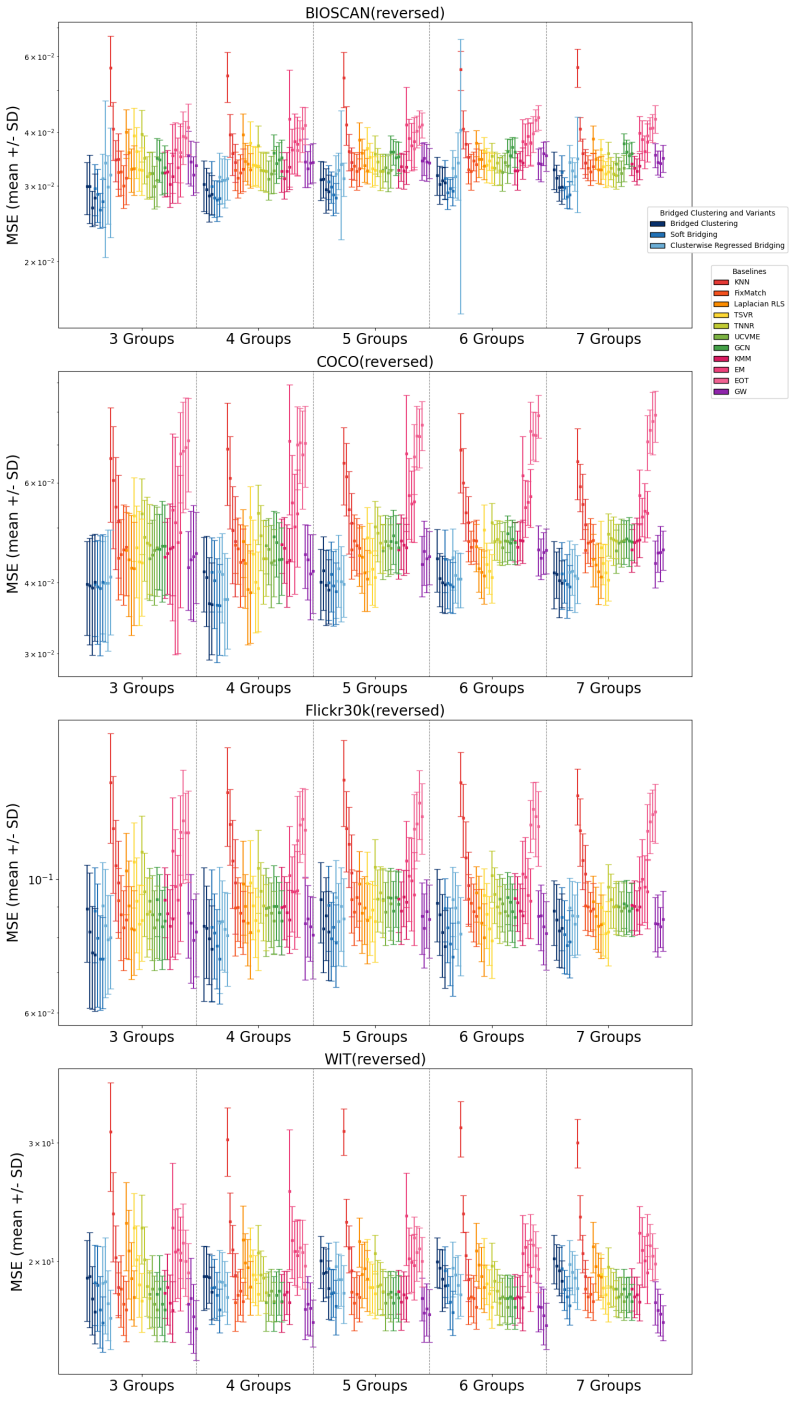


Figure 9: MSE distribution of different models in reversed experiments in the inductive setting. The 4 distribution plots of the same color represent the settings with 1,2,3,4 supervised samples per cluster.

F Detailed Experiment Results: Cluster Quality and Accuracy

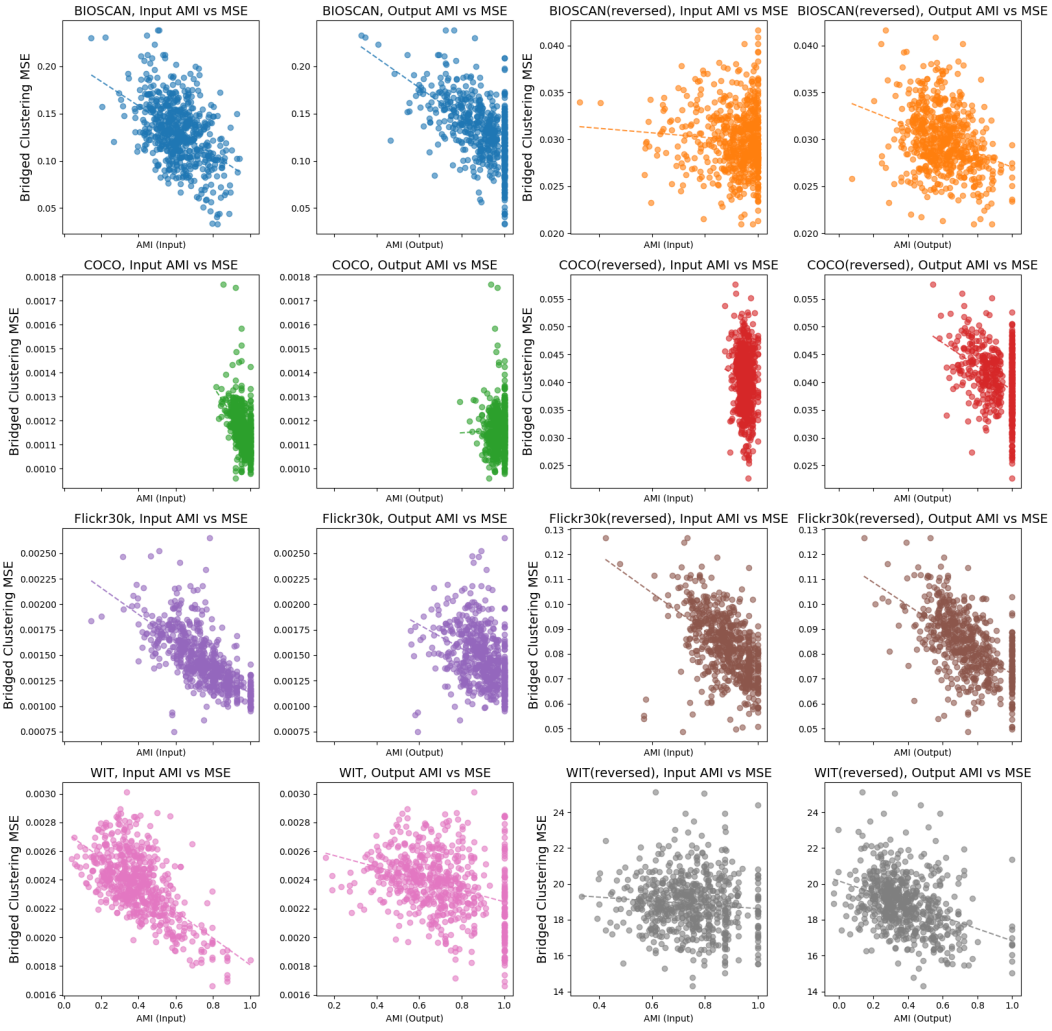


Figure 10: Mean Squared Error of Bridged Clustering V.S. Adjusted Mutual Information of Clusters (Transductive)

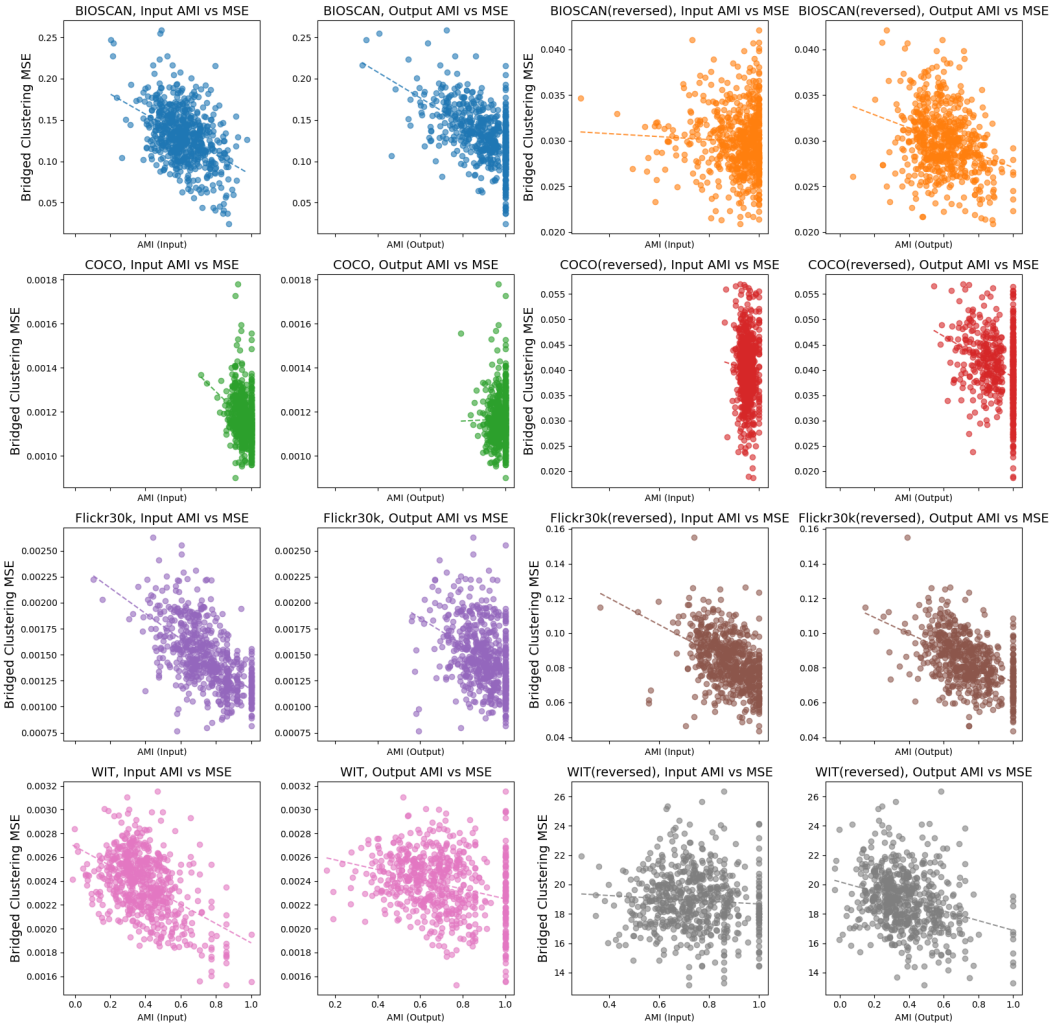


Figure 11: Mean Squared Error of BRIDGED CLUSTERING V.S. Adjusted Mutual Information of Clusters (Inductive)

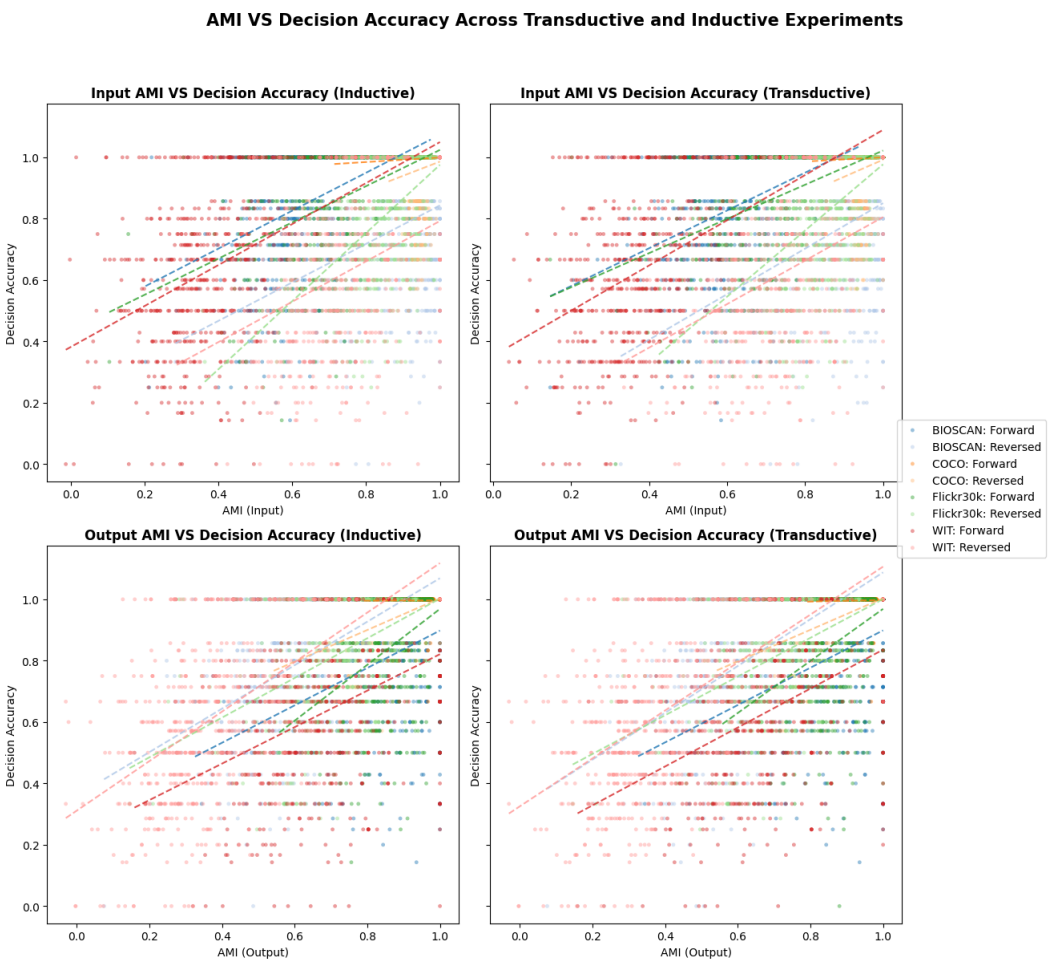


Figure 12: Bridging Accuracy as function of Adjusted Mutual Information of input/output clusters.

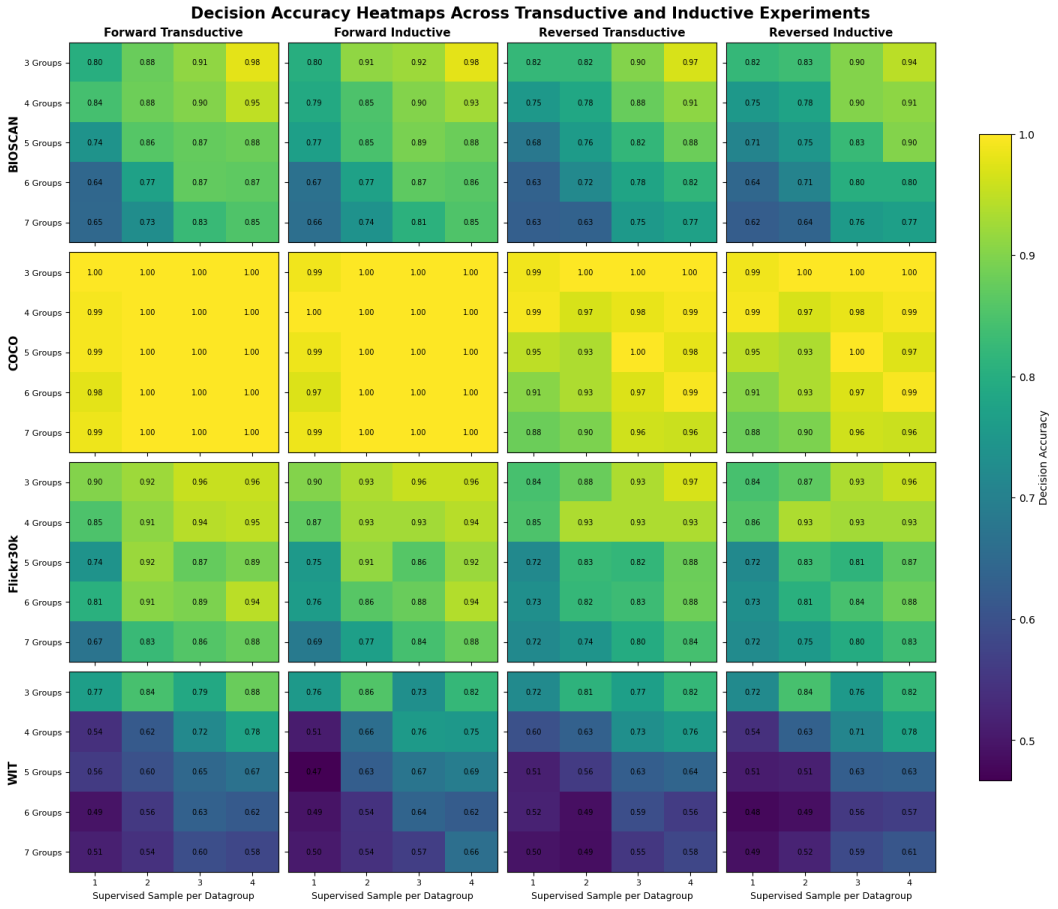


Figure 13: Average Bridging Decision Accuracy across Experiments

G EXPERIMENT RESULTS FOR ADVERSARIAL CONDITIONS

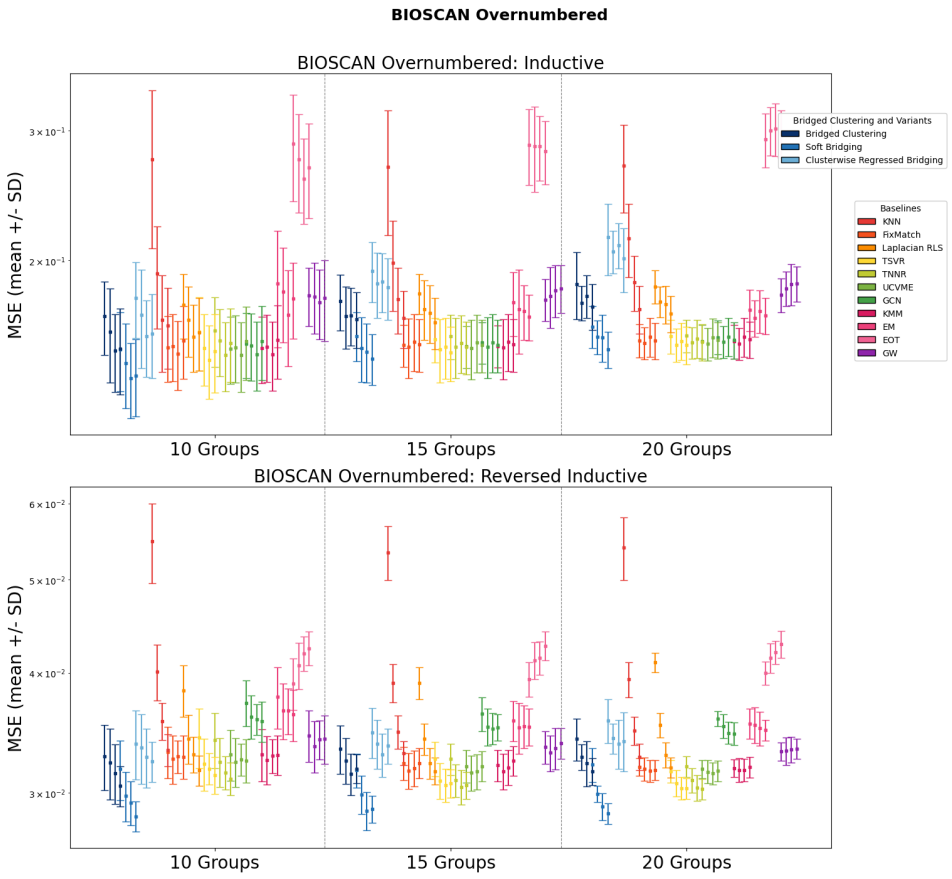


Figure 14: Running the main experiment on large cluster counts for BIOSCAN. We test only the inductive setting because this experiment is computationally expensive.

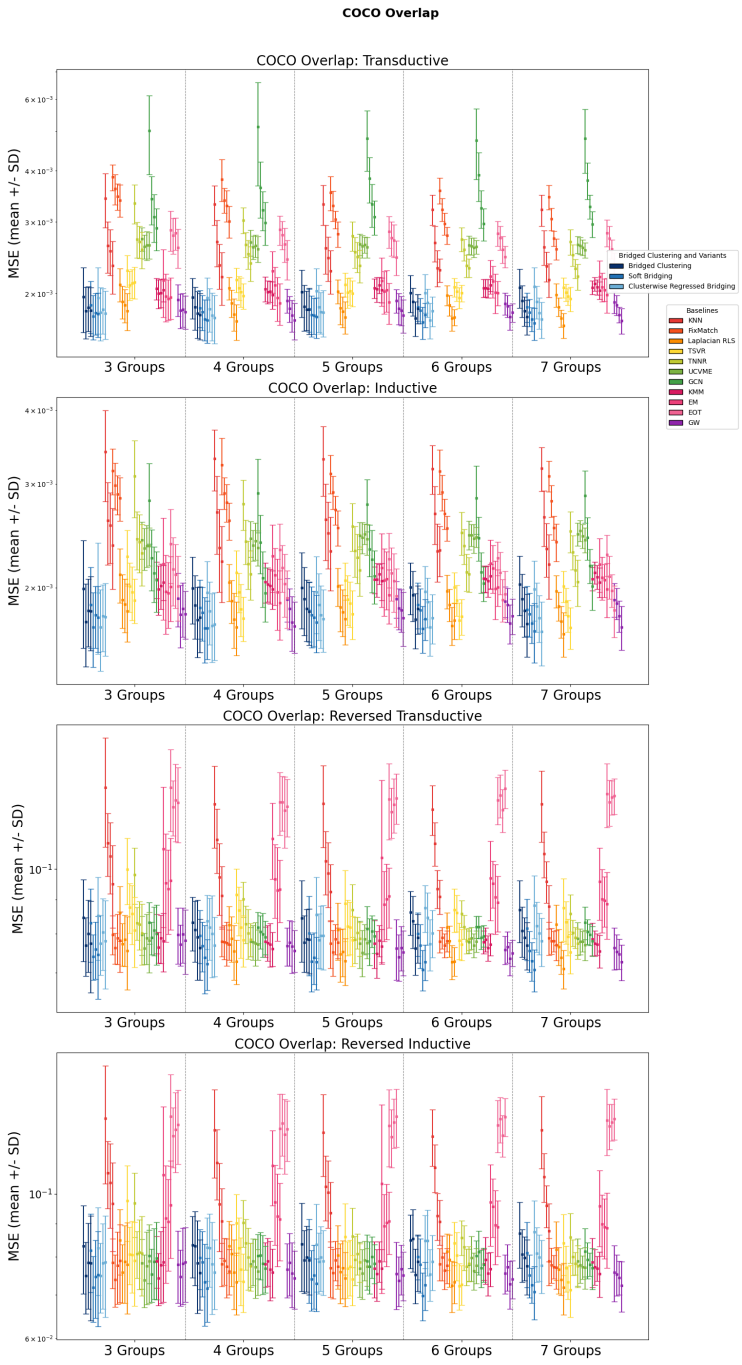


Figure 15: Running main experiment on COCO with Overlapping Data Classes

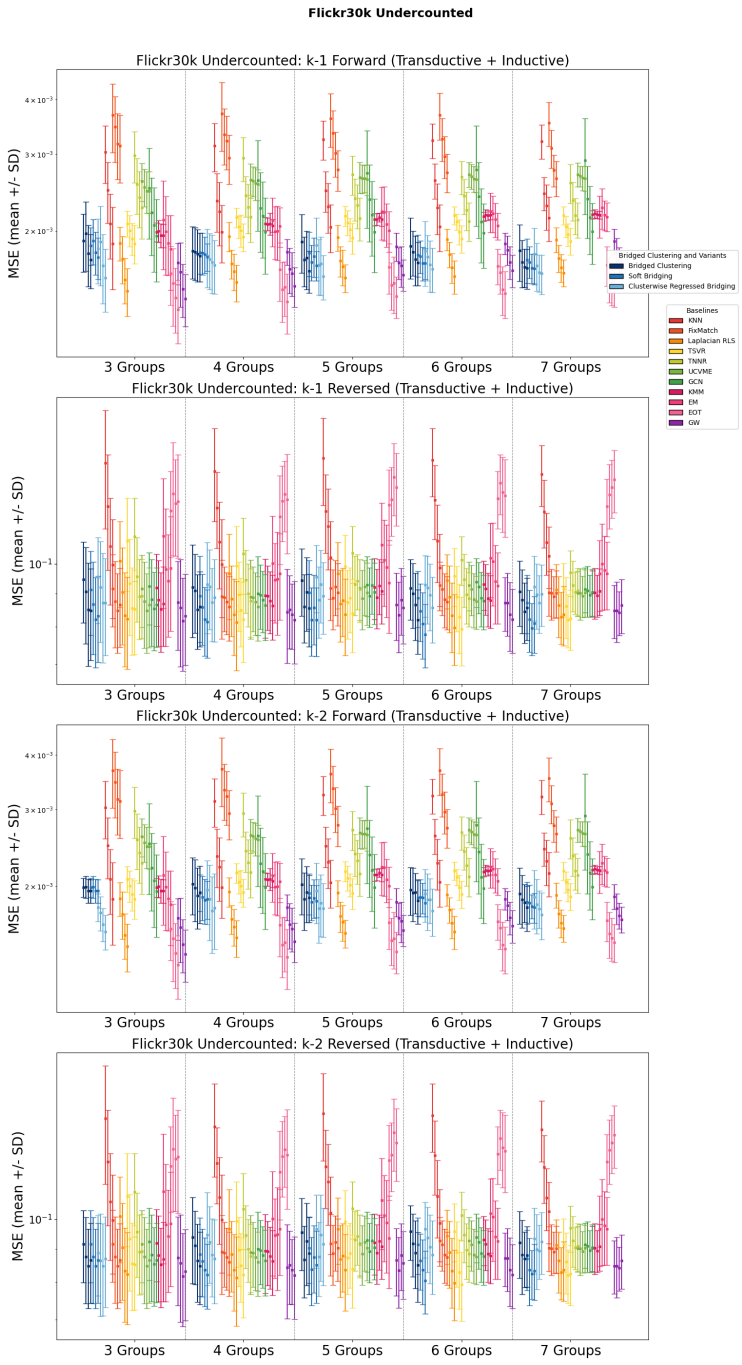


Figure 16: Running main experiment on undercounted cluster numbers for Flickr30k (averaged for transductive and inductive for visualization). Clustering algorithm deliberately undercounts cluster number by 1 and 2.

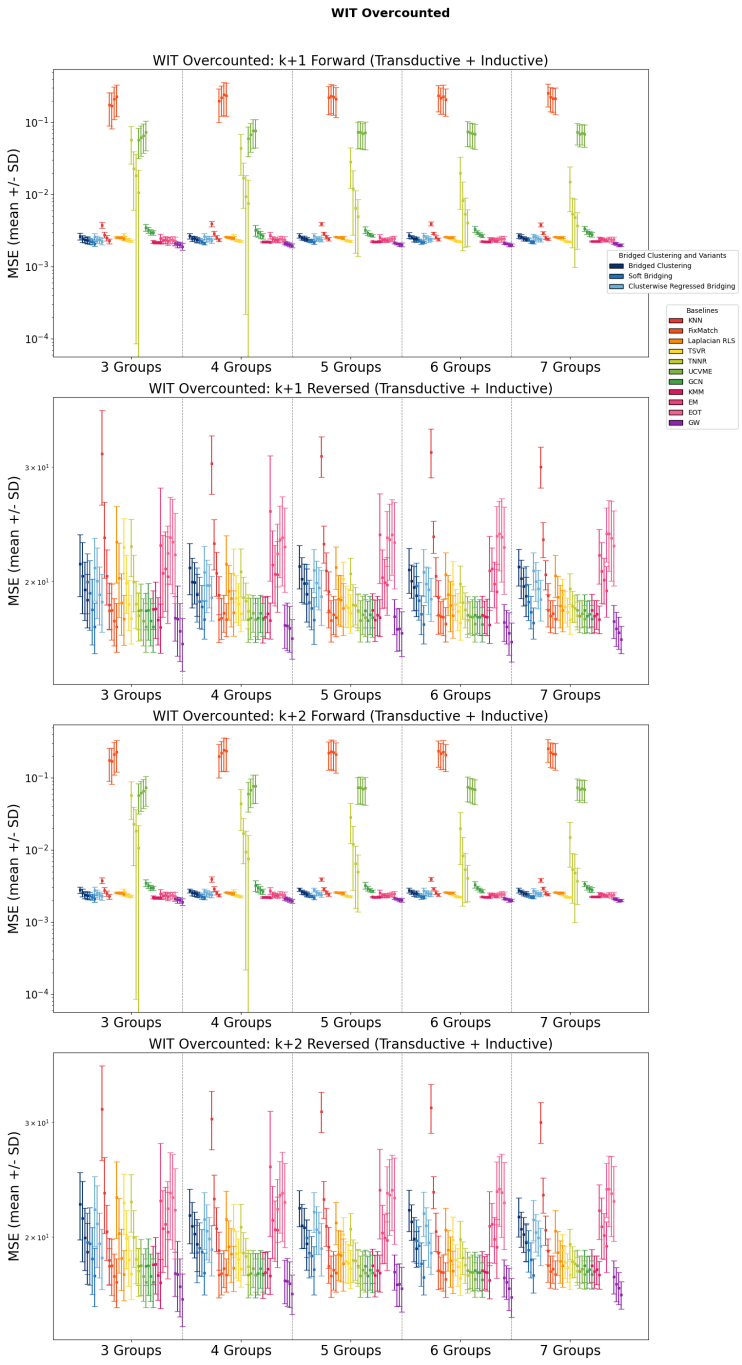


Figure 17: Running main experiment on overcounted cluster numbers for WIT (averaged for transductive and inductive for visualization). Clustering algorithm deliberately overcounts cluster number by 1 and 2.

H ILLUSTRATIONS FOR BRIDGED CLUSTERING

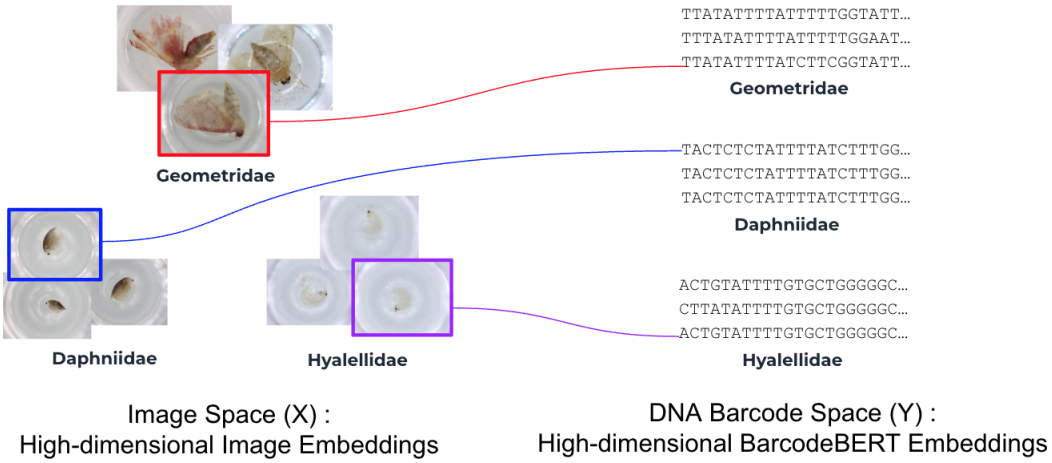


Figure 18: Illustrations for Bridged Clustering in BIOSCAN. In both Image and DNA spaces, datapoints from same species are often clustered together due to embedding proximity. Highlighted datapoints are supervised.

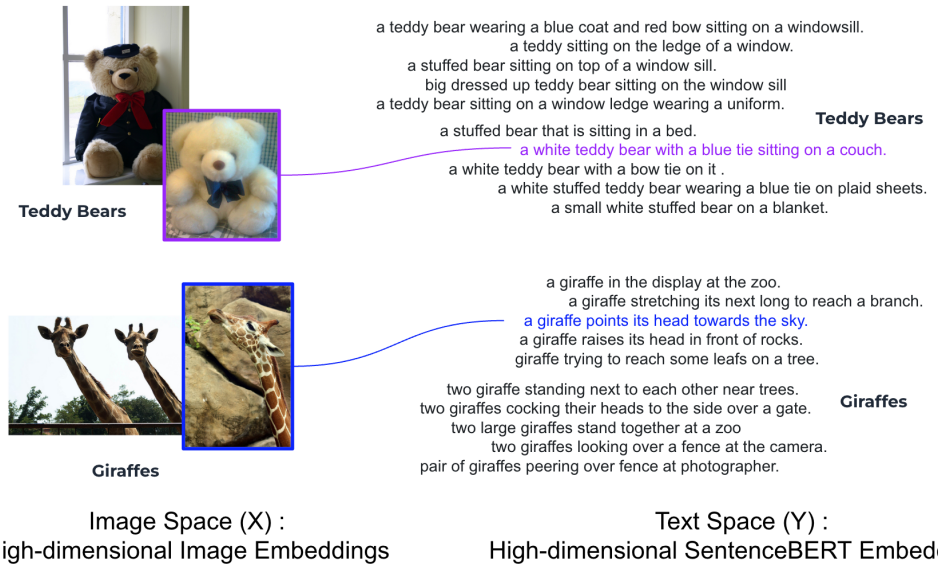


Figure 19: Illustrations for Bridged Clustering in COCO. In both Image and text spaces, datapoints from similar categories are often clustered together due to embedding proximity. Highlighted datapoints are supervised.

I ADDITIONAL INFORMATION

More reproducibility information about our computational experiments is available at our code repository, including method implementations, data processing, final hyperparameters, statistical tests and more.