Bridged Clustering for Computer Vision

Ellie Tanimura
etanim@stanford.edu

Pierre Labroche
labroche@stanford.edu

Abstract

We study the problem of semi-supervised learning in vision tasks with complex and heterogeneous output spaces. Standard methods typically assume aligned input-output structures or require dense supervision, limiting their applicability in cross-modal settings with sparse labels. Bridged Clustering (BC), a recently proposed framework, addresses this by independently clustering inputs and outputs, then using a small labeled set to learn a sparse alignment—or "bridge"—between the two spaces. In this work, we benchmark BC on two vision tasks: year prediction in WikiArt and ingredient detection in Food-101, demonstrating strong performance under minimal supervision. We further propose two extensions: Softmax-BC, which introduces soft label assignments, and GNN-BC, which propagates labels through an input-space graph to improve robustness. Across both tasks, GNN-BC consistently outperforms state-of-the-art baselines and BC variants, particularly in low-supervision regimes. Our results suggest that exploiting latent output structure via bridging and graph-based smoothing offers a powerful approach to semi-supervised learning in complex domains.

1. Introduction

Many real-world vision tasks suffer from prohibitively high labeling costs due to the need for expert annotations, expensive data collection, or specialized equipment. Yet, these domains often possess large amounts of unlabeled data.

Semi-supervised learning (SSL) attempts to bridge the gap between difficult-to-obtain labeled data and abundant unlabeled data by using structure in the input space to improve learning. Most SSL approaches focus on organizing or augmenting the input feature space so that unlabeled inputs can help shape a more robust representation of the data distribution [12].

However, many tasks have complex or sparse output spaces that are poorly represented by a few available labeled examples. For instance, the label space in historical artwork classification requires nuanced stylistic categories. In such cases, unlabeled outputs exist in abundance (e.g., years, art movements, etc.) but are rarely incorporated into existing SSL methods. This motivates an important extension to standard SSL: leveraging structure in the output space as well.

In a previous project, members of the Vitercik Lab proposed the Bridged Clustering (BC) algorithm, which independently clusters the input and output domains and uses a small set of labeled examples to learn a sparse alignment—or "bridge"—between the clusters. By clustering both inputs and outputs and learning how they align using only a few labeled examples, BC hopes to amplify the signal of sparse labels and better generalize across complex data distributions. BC is especially appealing for application in cross-modal prediction tasks because they typically do not require dense alignment between input and output clusters X and Y; instead, assuming that both spaces share latent structure.

While promising, BC has only been evaluated on a small toy dataset, against simple baselines such as K-Nearest Neighbors. In this project, we aim to benchmark and extend the BC algorithm to better understand its potential in semi-supervised vision tasks.

We evaluate BC across two different semi-supervised vision tasks: year prediction in WikiArt (scalar regression), and ingredient detection in food images (multi-label classification). These tasks differ in output structure and supervision sparsity, offering robust benchmarks for BC.

To enhance Bridged Clustering (BC), we propose several extensions that increase its flexibility and performance. Drawing inspiration from recent advances in graph-based SSL, we introduce two directions: cluster-aware smoothing using Graph Neural Networks (GNNs) and soft-label bridging. These extensions aim to improve predictive accuracy, particularly in cases involving ambiguous or imbalanced clusters.

2. Related Works

Most SSL methods in computer vision concentrate on shaping the input space using unlabeled data. Consistency-based approaches like Temporal Ensembling [7] and

Mean Teacher [11] encourage models to produce stable predictions under perturbations or temporal smoothing. Graph-based methods such as Laplacian Regularized Least Squares (LapRLS) [15] and GNNs [8] propagate labels across a learned data manifold, encouraging local smoothness and cluster coherence.

These methods assume strong alignment between inputs and outputs. Therefore, they are less effective in scenarios where labels are sparse or structurally different from the input domain, such as in cross-modal prediction. For such settings, methods like Balanced K-Means (BKM) [2], RankUp [4], and Unsupervised Clustering via Variational Manifold Embedding (UCVME) [3] improve generalization in settings by preserving the geometric or relational properties of the data, including clustering balance, ranking order, and manifold structure respectively. Twin Support Vector Regression (TSVR) [5] and Transductive Nearest Neighbor Regression (TNNR)[13] further extend this idea through transductive regimes where they excel at leveraging limited labeled data to infer smooth predictions over the entire dataset by learning dual regression hyperplanes in a semi-supervised, transductive setting and propagation of continuous labels through a nearest-neighbors graph using a small labeled set. Still, these approaches suffer in situations where the inputs and outputs differ significantly in modality or dimensionality.

On the other hand, BC is well-suited to these transductive scenarios. By clustering inputs and outputs separately and learning a sparse alignment via a small labeled set, BC sidesteps the need for aligned representations or dense supervision. Rather than predicting exact labels, BC assigns an input to a cluster, maps it to an output cluster using the learned bridge, and returns the centroid of the output cluster as the prediction.

Additionally, recent work has shown that GNNs can effectively propagate labels and smooth out cluster assignments across data manifolds [6]. We build on this insight to extend BC with soft bridging and GNN-based smoothing, allowing for more flexible cluster assignment and improved handling of noise and ambiguity in real-world datasets.

3. Problem Statement

Let $\{(x_i, y_i)\}_{i=1}^n$ be a dataset where the majority of data is unlabeled, and a small subset $\mathcal{S} \subset \{(x_i, y_i)\}$ is labeled. Bridged Clustering assumes that both the input space $\mathcal{X}$ and the output space $\mathcal{Y}$ admit latent cluster structure.

The core objective is to leverage this structure by:

1. Clustering the input and output spaces independently using an unsupervised algorithm.

2. Using the small supervised set $\mathcal{S}$ to learn a bridge—a sparse alignment—between input and output clusters.

3. Predicting outputs for unlabeled inputs by mapping their cluster assignments in the input space to corresponding output clusters, and using the associated cluster centroids as predictions.

We consider both hard and soft variants of this framework, which differ in how cluster assignments and predictions are computed. These variants are described in detail in the following section.

4. Datasets

We evaluate Bridged Clustering across two datasets spanning scalar regression and multi-label classification.

WikiArt. The WikiArt dataset contains over 80,000 paintings sourced from WikiArt.org, annotated with metadata including creation year and artistic style [10]. The dataset spans works by $1,119$ artists and is categorized into 27 distinct artistic styles. We treat the year as a scalar regression target and standardize it across the dataset. To induce meaningful cluster structure, we filter the original set of images to include only those corresponding to 2 to 6 randomly selected artistic styles per trial. The assumption is that artistic style serves as a latent factor shared between the input space (images) and the output space (year), allowing Bridged Clustering to discover structure even without explicit supervision over style. See Figure 1 for an example of an image in WikiArt dataset.

Figure 1. Example WikiArt image.

"Food-101": Ingredient Extraction. This dataset consists of over $4,000$ food images annotated with ingredient lists [9]. We binarize each list into a multi-hot vector over the 1,095 most frequent ingredients and frame the task as multi-label prediction. In each trial, we select images from 2 to 6 distinct food categories with the goal of inducing latent cluster structure shared between the input space (images) and the output space (ingredients). The assumption is that cuisine acts as a common underlying factor that organizes both the visual features and the ingredient distributions, enabling Bridged Clustering to align input and output clusters without access to category labels. See Figure 2 for an example of an image in the Food-101 dataset.

Figure 2. Example "Food-101" cuisine image.

Preprocessing. Images from both datasets were resized to 224 × 224 pixels and z-score normalized using channel-wise means of $[0.486, 0.456, 0.406]$ and standard deviations of $[0.229, 0.224, 0.225]$.

5. Method

We implement the Bridged Clustering algorithm as originally proposed by Ye et al. [14] and propose novel variants that improve cluster assignment through label propagation, voting-based refinement, and task-specific input-output adaptations.

The algorithm consists of three main stages, which we detail over the following subsections.

5.1. Simulating the Bridged Clustering Setting

To reflect the practical conditions Bridged Clustering is designed for—where labeled input-output pairs are scarce—we partition each our dataset into three disjoint subsets: a small supervised set $\mathcal{S}$, an unlabeled input set $\mathcal{X}_{\text{unlabeled}}$, and an unlabeled output set $\mathcal{Y}_{\text{unlabeled}}$. The supervised set includes just 1, 3, 5, or 10 labeled examples per output cluster, simulating the extremely low-resource regimes that are common in vision-based semi-supervised learning.

The remaining data is split between $\mathcal{X}_{\text{unlabeled}}$ and $\mathcal{Y}_{\text{unlabeled}}$. The specific division is an ablated hyperparameter that we will discuss in the Results section. This disjoint sampling allows us to determine whether a meaningful alignment can be learned between latent input and output structures even when they are not observed together. A schematic overview of this partitioning is shown in Figure 3, where "Unsupervised Set 1" corresponds to $\mathcal{X}_{\text{unlabeled}}$ and "Unsupervised Set 2" to $\mathcal{Y}_{\text{unlabeled}}$.

5.2. Clustering and Representation

After splitting the dataset, we use a pretrained ResNet-50 model to produce 2048-dimensional input feature vectors x_i from the entries in $\mathcal{X}_{\text{unlabeled}}$. Output labels $y_i \in \mathcal{Y}_{\text{unlabeled}}$ are either scalar values (WikiArt) or 1095-dimensional binary vectors (Food-101). To reduce computation cost during clustering in Food-101, we apply a Gaussian random projection to the 1095-dimensional binary label vectors, reducing them to 128-dimensional real-valued vectors. We

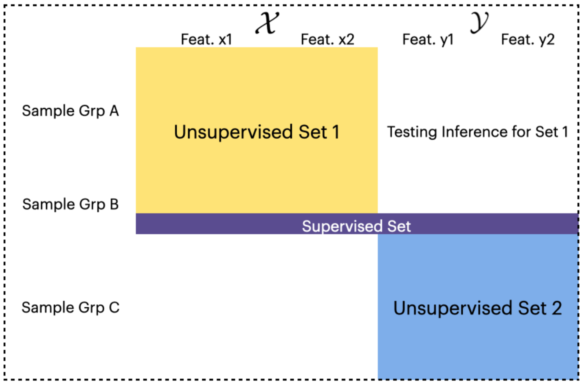

Figure 3. Dataset subdivision for Bridged Clustering simulation

cluster the input space $\mathcal{X}$ and output space $\mathcal{Y}$ independently using K-Means with k clusters. Let:

- $C_X : \mathcal{X} \to \{1, \dots, k\}$ denote input cluster assignments,
- $C_Y : \mathcal{Y} \to \{1, \dots, k\}$ denote output cluster assignments.

5.3. Bridge Learning

Using the small supervised set $\mathcal{S} = \{(x_i, y_i)\}_{i=1}^{s}$, we estimate a mapping $A : \{1, \dots, k\} \to \{1, \dots, k\}$ from input to output clusters.

Hard Alignment with Majority Voting. In the hard alignment setting, we define A using majority voting [1]:

$$A(c) = \arg\max_{c'} \sum_{(x_i, y_i) \in \mathcal{S}} \mathbf{1}\{C_X(x_i) = c \text{ and } C_Y(y_i) = c'\}$$

Then, we define $\hat{y}(x_j)$ for each test input x_j as the centroid of the mapped output cluster:

$$\hat{y}(x_j) = \mu_{A(C_X(x_j))}$$

where $\mu_{c'}$ denotes the centroid of output cluster c' in the output space.

Hard Alignment with Hungarian Voting. We define a "co-occurrence" matrix:

$$M_{c,c'} = \sum_{(x_i, y_i) \in \mathcal{S}} \mathbf{1}\{C_X(x_i) = c \text{ and } C_Y(y_i) = c'\}$$

for $c, c' = 1, \dots, k$.

Then, $A_{Hungarian}$ is the mapping that maximizes:

$$A_{Hungarian} = \arg\max_{\pi} \sum_{(x_i, y_i) \in \mathcal{S}} M_{c,\pi(c)}$$

where π is a permutation of $\{1, \dots, k\} \to \{1, \dots, k\}$.

The above is typically solved by applying the Hungarian (e.g., linear sum assignment) algorithm to the cost matrix $-M$. Once solved, we define our prediction for each test input x_j as:

$$\hat{y}^{Hungarian}(x_j) = \mu_{A_{Hungarian}}(C_X(x_j))$$

Softmax Alignment. Note: for notational clarity, the following alignments use the mapping A and corresponding predictions $\hat{y}$ obtained via hard alignment with majority voting. However, these can be readily substituted with those from hard alignment using Hungarian voting, without loss of generality.

We propose a soft-label variant where each input softly votes across output clusters based on distance-weighted similarity, and predictions are computed as a weighted average of output centroids. This approach is motivated by the intuition that some inputs may lie near the boundary between multiple clusters, and a hard assignment may discard useful uncertainty information. By incorporating a soft distribution over clusters, we hope to produce smoother and potentially more robust predictions in ambiguous or noisy regions of the input space.

First, we compute the hard-aligned prediction from the original Bridged Clustering algorithm:

$$\hat{y}(x_j) = \mu_{A(C_X(x_j))}$$

Next, we define an inverse-distance softmax over all k output centroids:

$$p_i(x_j) = \frac{\exp\left(-\|\hat{y}(x_j) - \mu_i\|_2\right)}{\sum_{\ell=1}^{k} \exp\left(-\|\hat{y}(x_j) - \mu_\ell\|_2\right)} \quad \text{for } i = 1, \ldots, k$$

Finally, we compute the smoothed prediction as the expected output under this distribution:

$$\hat{y}_j^{\text{softmax}} = \sum_{i=1}^{k} p_i(x_j) \cdot \mu_i$$

While conceptually appealing, this approach underperforms compared to hard alignment in low-supervision settings. We hypothesize that averaging across uncertain cluster assignments causes predictions to collapse toward the mean, obscuring fine-grained structure. This motivates our GNN-based extension, which retains local structure without excessive smoothing.

GNN Alignment. To incorporate geometric structure in the input space, we construct a k-nearest-neighbor (k-NN) graph over the input features $X \in \mathbb{R}^{N \times D_x}$, where:

- N is the number of input samples (from $\mathcal{X}_{\text{unlabeled}} \cup \mathcal{S}$),
- $D_x = 2048$ is the dimensionality of the input features (e.g., ResNet-50 embeddings),
- $X = [x_1, \ldots, x_N]^\top$ is the matrix of input embeddings.

We define an adjacency matrix $A \in \{0,1\}^{N \times N}$ for the k-NN graph and compute its symmetrically normalized form $\tilde{A}$.

A three-layer Graph Convolutional Network (GCN) is then trained to predict output cluster assignments on the supervised examples. The GCN operates as follows:

$$H^{(1)} = \text{ReLU}(\tilde{A}XW^{(0)}) \tag{1}$$

$$H^{(2)} = \text{ReLU}(\tilde{A}H^{(1)}W^{(1)}) \tag{2}$$

$$P = \text{softmax}(\tilde{A}H^{(2)}W^{(2)}) \in \mathbb{R}^{N \times k} \tag{3}$$

where:

- $W^{(0)} \in \mathbb{R}^{D_x \times d}$, $W^{(1)} \in \mathbb{R}^{d \times d}$, and $W^{(2)} \in \mathbb{R}^{d \times k}$ are learnable weight matrices,
- d is the hidden dimensionality of the intermediate GCN layer,
- $P_i = [p_{i1}, \ldots, p_{ik}]$ denotes the predicted distribution over output clusters for input x_i.

The model is trained using cross-entropy loss on the supervised subset $\mathcal{S}$, where ground-truth output cluster labels are given by $C_Y(y_i)$.

At inference time, the prediction for input x_i is computed as the expectation over output centroids:

$$\hat{y}^{\text{GNN}}(x_i) = \sum_{j=1}^{k} p_{ij} \cdot \mu_j$$

This GNN-Bridge variant allows the model to smooth noisy or ambiguous input assignments by propagating supervision through the input graph, while still collapsing to hard-aligned behavior when confident.

6. Experiments

Experimental Goals. Our experiments are designed to evaluate the performance of Bridged Clustering (BC) under low-supervision regimes and to understand how alignment and smoothing strategies affect its predictive accuracy. We compare BC against a range of supervised and semi-supervised baselines, investigate the effect of different voting schemes for cluster alignment, and assess whether our proposed extensions—Softmax-BC and GNN-BC—offer improvements, particularly in noisy or ambiguous cluster settings. We also perform a targeted hyperparameter search to optimize the GNN variant and analyze how model behavior changes across cluster granularity and supervision levels.

Experimental Setup. We evaluate each task across cluster counts $k \in \{2,3,4,5,6\}$, with additional results for $k = 8, 10$ included in the Appendix, which support similar conclusions. Since cluster boundaries are unsupervised and may not align with semantic classes (e.g., artistic styles or cuisines), varying k allows us to test robustness to different structural assumptions. Lower values of k capture coarse groupings (e.g., broad historical periods or major ingredient types), while higher values introduce finer distinctions. For each k, we construct a dataset by randomly sampling k classes from the full label set (25 in WikiArt, 101 in Food-101), ensuring diverse class combinations across trials. We treat $k = 6$ as a meaningful upper bound in our main experiments, as it often includes semantically overlapping or visually similar categories—such as Renaissance substyles or regional cuisines—that pose greater alignment challenges without introducing excessive fragmentation.

Supervision is limited to 1, 3, 5, or 10 labeled examples per output cluster, simulating realistic low-resource conditions where annotations are costly or sparse. This reflects many domain-specific applications in which generalization from minimal labeled data is required.

We run each configuration (i.e., a combination of k, supervision level, and randomly sampled data subset) is over 100 trials for stability.

Baselines. We compare Bridged Clustering and our proposed variants against seven state-of-the-art baselines: K-Nearest Neighbors (KNN), XGBoost, Mean Teacher, Laplacian Regularized Least Squares (LapRLS), Transductive Support Vector Regression (TSVR), Uncertainty-Consistent Variational Model Ensembling (UCVME), and RankUp. Each model is configured for its respective output domain (scalar or multi-label) and trained solely on the supervised subset.

Hyperparameter Ablation. To assess the sensitivity of Bridged Clustering (BC) and its extensions, we ablate two core components: the voting scheme for input-output alignment, and the hyperparameters for GNN-BC.

Voting Scheme. We compare majority voting (many-to-one cluster mapping) with Hungarian voting (one-to-one assignment) to evaluate the effect of alignment granularity. Both schemes are tested across datasets, supervision levels, and cluster counts; performance differences are discussed in the Results section.

GNN-BC Tuning. Guided by prior work in semi-supervised graph learning [6], we sweep learning rates from 1e^{-5} to 9e^{-3}, test weight decay values 1e^{-5} and 1e^{-4}, and ablate hidden dimensions $\{64, 128, 256, 512\}$, with 128 and 256 commonly used in practice. We vary the k-NN graph size over $k \in [1, 50]$ and cap training at 1000 epochs for convergence. The best setup—used in all GNN-BC evaluations—uses the Adam optimizer, learning rate 3×10^{-3}, weight decay 1×10^{-5}, hidden dimension 128, $k = 22$, and 47 training epochs.

We evaluate each combination of hyperparameters using an 80/20 split of the supervised set into train and validation data.

Evaluation Metric. We use mean absolute error (MAE) as our primary metric due to its interpretability across both scalar and multi-label output spaces, and its robustness to outliers. For each dataset and configuration, we report the average MAE across 100 trials. Note that the MAEs are not directly comparable between tasks: WikiArt involves year prediction, where errors can span centuries, while Food-101 uses a 0–1 multi-hot vector over 1,095 ingredients.

Qualitative and Quantitative Analysis. In addition to MAE, we present cluster visualizations (Figures 7–9) and task-specific error breakdowns. Common failure modes, such as cluster collapse in Softmax-BC or over-smoothing in GNN-BC, are discussed in the Results section.

7. Results and Analysis

We report results for both WikiArt (scalar regression) and Food-101 (multi-label classification) tasks. For each dataset, we compare Bridged Clustering and its variants (Softmax-BC, GNN-BC) against a suite of baselines across multiple cluster settings and supervision levels. Unless otherwise noted, results represent the mean absolute error (MAE), averaged across 100 trials.

7.1. WikiArt: Year Prediction

BC vs Baselines. BC consistently outperforms classical baselines such as KNN and XGBoost across all cluster and supervision settings (Figure 4). These baselines fail to benefit meaningfully from added supervision, especially in sparse regimes, likely due to their inability to leverage unlabeled data. In contrast, BC uses the latent alignment between input and output clusters to generalize from minimal labeled data.

GNN-BC and Softmax-BC. GNN-BC improves upon BC in most settings, with performance gains increasing as supervision increases. Notably, statistical significance (Figure 5) becomes stronger at 3, 5, and 10 samples per cluster. We attribute this to the fact that with more labeled anchors, GNN-BC can construct more reliable neighborhood graphs, enabling effective label propagation across structurally similar but unlabeled points.

At extremely low supervision (e.g., 1 sample/cluster), GNN-BC performs comparably or slightly worse than BC.

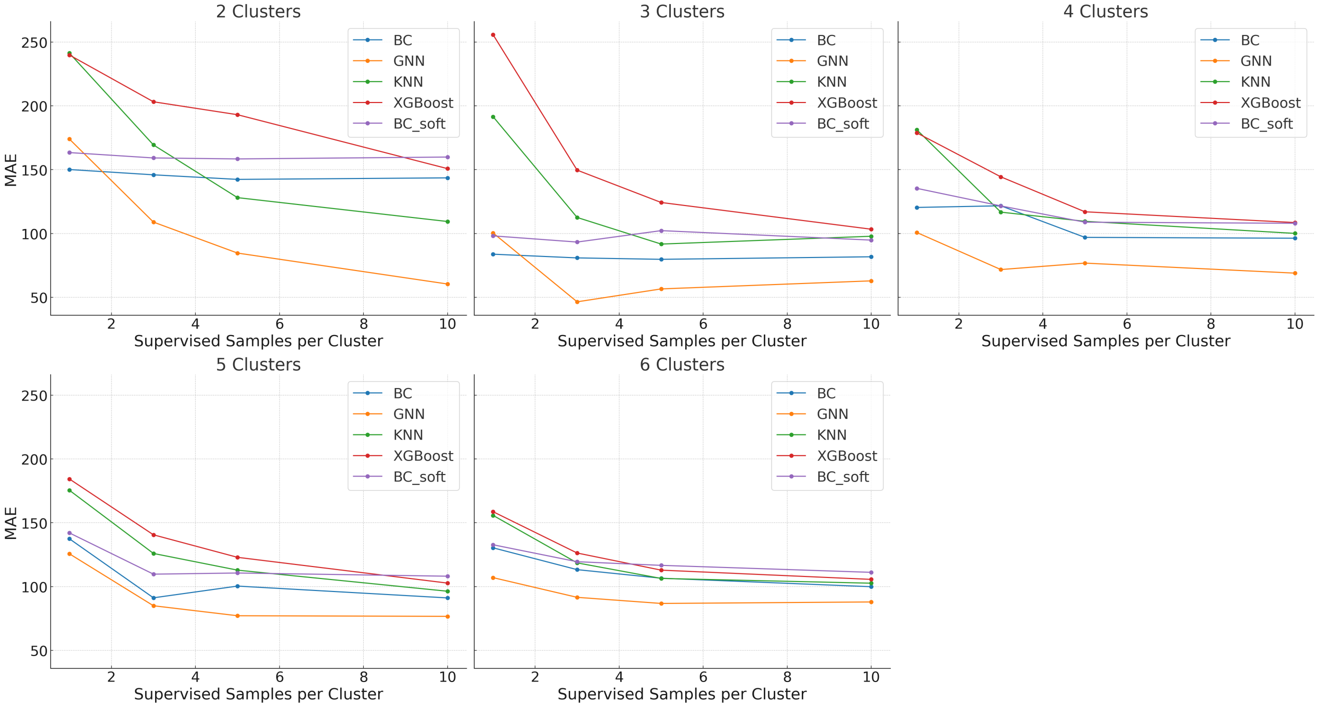

Figure 4. Line plots of MAE vs. supervision level for each method across different cluster sizes k. Each point reflects the average over 100 trials.

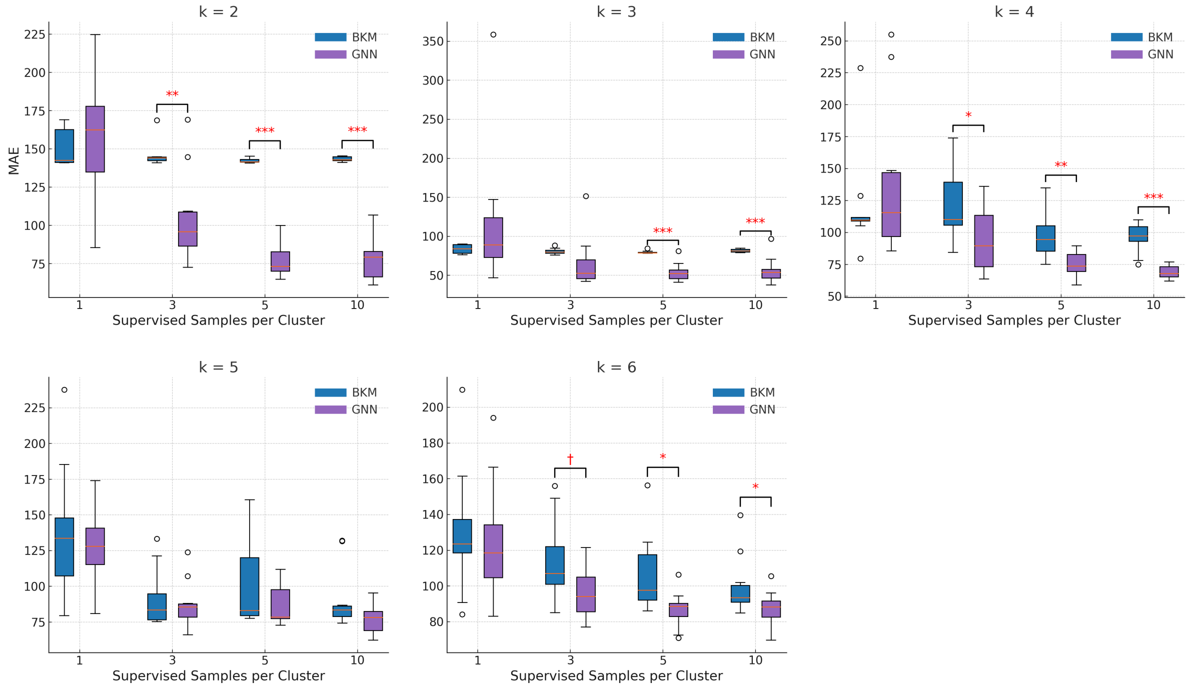

Figure 5. MAE comparison between GNN-BC and BC (labeled BKM) across cluster sizes $k \in \{2, 3, 4, 5, 6\}$ on WikiArt. Statistical significance is indicated by paired t-tests over 100 trials: †: $p < 0.1$, *: $p < 0.05$, **: $p < 0.01$, ***: $p < 0.001$.

We believe this is due to oversmoothing: in the absence of sufficient labeled anchors, label information diffuses across the graph indiscriminately, which in turn reduces the model's ability to preserve class-specific distinctions. This effect is more pronounced when graphs contain noisy or loosely connected components, which misalign input-output mappings and lead to prediction drift.

Softmax-BC performs worst across all settings. Its MAE curves are flat and consistently higher than BC and GNN-BC. We attribute this to softmax collapse, where soft weighting over cluster centers causes predictions to gravitate toward the global mean. Our target (year) spans a wide numeric range, and the unnormalized L1 distances between cluster centroids are often large in magnitude, leading to overly sharp or flat softmax distributions depending on scale. A potential remedy is to normalize distances prior to softmax, which would equalize the influence of each centroid and reduce collapse.

Effect of Cluster Size. All methods improve with increasing supervision, but show diminishing returns beyond 5 samples per cluster. GNN-BC benefits the most from additional supervision, particularly at $k = 3$ to 5, where graph quality and label coverage reach a sweet spot. Too few clusters (e.g., $k = 2$) result in overly coarse partitions, limiting the resolution of learned structure. Too many clusters (e.g., $k = 6$) can fragment the input space, leading to unstable bridges and noisier predictions. Overall, $k = 4$ and $k = 5$ offer the best trade-off between semantic granularity and alignment stability.

Summary. BC and its variants outperform all baselines on WikiArt, with GNN-BC achieving the best results when given sufficient supervision. Its gains stem from effective structural smoothing and graph-based propagation. In contrast, Softmax-BC fails to leverage supervision or structure due to averaging effects that obscure fine-grained targets. These results highlight the importance of explicit structure-preserving mechanisms in sparse regression tasks.

7.2. Food-101: Cuisine Classification

Voting Strategy. We compare majority voting and Hungarian voting across 10 trials per cluster-supervision setting. Majority voting consistently outperforms, likely because it permits many-to-one mappings between input and output clusters. This flexibility is crucial when multiple visual clusters correspond to a single cuisine label, which is common in noisy real-world data.

BC vs Baselines. As shown in Figure 6, BC consistently outperforms KNN and LapRLS across all configurations. Unlike the baselines, which rely solely on the limited labeled set, BC leverages unlabeled data through structural alignment. This results in more robust generalization, especially under sparse supervision. The closeness in performance between BC and its variants also suggests that the learned bridge captures strong semantic alignment between input clusters and output cuisines.

GNN-BC and Softmax-BC. GNN-BC performs similarly to BC but generally underperforms slightly. Its dependence on a KNN graph makes it vulnerable to misalignment when input features do not reflect semantic similarity. Additionally, GNN-BC requires more labeled anchors to propagate meaningful signals; with limited supervision, it fails to learn stable weights, leading to poor generalization.

Softmax-BC trails both BC and GNN-BC in all settings. Because it computes a weighted average of output cluster centroids, even small affinities to incorrect clusters drag predictions away from the true label. Without mechanisms for denoising or correction, it struggles to maintain accuracy in the presence of ambiguity or noise.

Effect of Cluster Size. As cluster size increases, BC and its variants continue to outperform the baselines, especially with more supervision. The bridge becomes more precise with additional labels, enabling better alignment between input features and output classes. In contrast, the baselines remain heavily constrained by their limited access to supervision.

Failure Modes. At $k = 2$, all methods perform similarly due to the simplicity of the clustering problem. As supervision increases, BC and its variants start to significantly outperform the baselines. This indicates that the bridge benefits most when the problem has moderate complexity and sufficient labels to stabilize alignment.

7.3. Cross-Task Comparison

The WikiArt and Food-101 results highlight key differences in how each BC variant behaves under different supervision and label structure conditions:

GNN excels on WikiArt due to the smooth, continuous nature of the year variable. Local neighborhood structure in the learned graph provides meaningful signals, enabling effective label propagation—especially as supervision increases. The benefits of GNN-BC grow with more labeled anchors, which allow it to correct for noise and avoid oversmoothing.

GNN-BC underperforms slightly on Food-101, where the output space is discrete and multi-label. Graph propagation is less effective because local neighbors may belong to different classes (e.g., visually similar dishes with distinct ingredients), leading to noise amplification. Furthermore, the supervision sparsity makes it difficult for GNN-BC to learn appropriate edge weights.

Softmax-BC underperforms on both tasks. Its averaging mechanism blurs class boundaries in classification and pulls predictions toward the mean in regression. This effect is especially problematic in WikiArt, where the wide numeric range of the year variable interacts poorly with unscaled softmax logits. Even in Food-101, where outputs are categorical, Softmax-BC cannot denoise or correct errors introduced by ambiguous input features.

Vanilla BC is the most consistent. It performs competitively across both datasets and often outperforms GNN-BC when supervision is very low. Its voting-based bridge sidesteps issues of noisy graph construction and avoids collapse from over-averaging.

Interestingly, on WikiArt, the NMI between image and year clusters remains stable at approximately 0.35 across all cluster sizes and supervision levels. A example of the clustering quality is shown in Figure 8. This suggests a relatively consistent but modest alignment between the visual representation of style (input) and temporal annotation (output). Despite this modest NMI, Bridged Clustering methods still achieve strong performance—particularly GNN-BC—by leveraging the local smoothness of the continuous year variable.

By contrast, Food-101 exhibits highly variable alignment between input (image) and output (ingredient) feature spaces, as shown in Figures 7 and 9. Some cuisines, like Red Velvet Cake or Caprese Salad (Figure 7), show strong cluster consistency between the two spaces, suggesting that GNN-based propagation could be effective in these cases. However, others, like Pizza and Sashimi (Figure 9), display poor alignment—with samples scattered across clusters in one space but concentrated in another.

This inconsistency likely degrades the overall performance of GNN-BC: while some regions of the graph may benefit from smooth label diffusion, others propagate conflicting signals, especially under sparse supervision. We think that although GNN-BC *can* exploit strong local structure in well-aligned regions, its performance is limited by the variability and unreliability of such structure across the dataset. In this sense, the graph becomes an inconsistent medium—sometimes helpful, but often harmful—ultimately making GNN-BC less robust for Food-101.

8. Conclusion

This work benchmarks and extends the Bridged Clustering (BC) algorithm for semi-supervised learning in vision tasks with complex output spaces. Our experiments across scalar regression (WikiArt) and multi-label classification (Food-101) demonstrate that BC—and particularly our GNN-augmented variant—can outperform or match a range of strong baselines under extremely low supervision.

We find that majority voting provides more robust cluster alignments than Hungarian voting, likely due to its flexibility in supporting many-to-one mappings between input and output clusters. Although Softmax-BC is conceptually appealing for its ability to represent uncertainty, it underperforms due to prediction collapse in sparse supervision regimes. In contrast, GNN-BC consistently improves performance by leveraging local neighborhood structure in the input space to propagate labels across ambiguous or noisy regions.

Overall, these results suggest that incorporating latent output structure can meaningfully enhance predictive accuracy in low-label regimes. The BC framework's modular design and ability to align cross-modal spaces make it especially well-suited for tasks where input and output modalities differ in nature or dimensionality.

Future Directions. Given the success of cluster-level bridging, future work could explore incorporating probabilistic or Bayesian formulations of the bridge itself. This would allow the model to express uncertainty over input-output mappings, which may be especially beneficial in settings with high output ambiguity or weak alignment—such as multi-label tasks like Food-101. Learning a distribution over possible bridges, rather than committing to a single deterministic mapping, could enable more robust inference in regions where supervision is sparse or conflicting.

9. Contributions and Acknowledgements

Ellie worked on the WikiArt dataset, and Pierre worked on the Food-101 dataset. Patrick Ye, the advisor for this community project, contributed to the initial skeleton code of the original Bridge Clustering (BC) algorithm and overall mentorship. Ellie and Pierre contributed to the analysis of their respective datasets, along with tuning and adjusting the baselines, BC, and its variants to the proper inputs and outputs.

References

[1] S. Aeeneh, N. Zlatanov, and J. Yu. New bounds on the accuracy of majority voting for multiclass classification. *IEEE Transactions on Neural Networks and Learning Systems*, 2024. 3

[2] X. Chang, F. Nie, Z. Ma, and Y. Yang. Balanced k-means and min-cut clustering. *arXiv preprint arXiv:1411.6235*, 2014. 2

[3] W. Dai, X. Li, and K.-T. 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, pages 7304–7313, 2023. 2

[4] P.-Y. Huang, S.-W. Fu, and Y. Tsao. Rankup: Boosting semi-supervised regression with an auxiliary ranking classifier. *arXiv preprint arXiv:2410.22124*, 2024. 2

[5] Jayadeva, R. Khemchandani, and S. Chandra. Twin support vector machines for pattern classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(5):905–910, 2007. 2

[6] T. N. Kipf and M. Welling. Semi-Supervised Classification with Graph Convolutional Networks, Feb. 2017. arXiv:1609.02907 [cs]. 2, 5

[7] S. Laine and T. Aila. Temporal ensembling for semi-supervised learning. *CoRR*, abs/1610.02242, 2016. 1

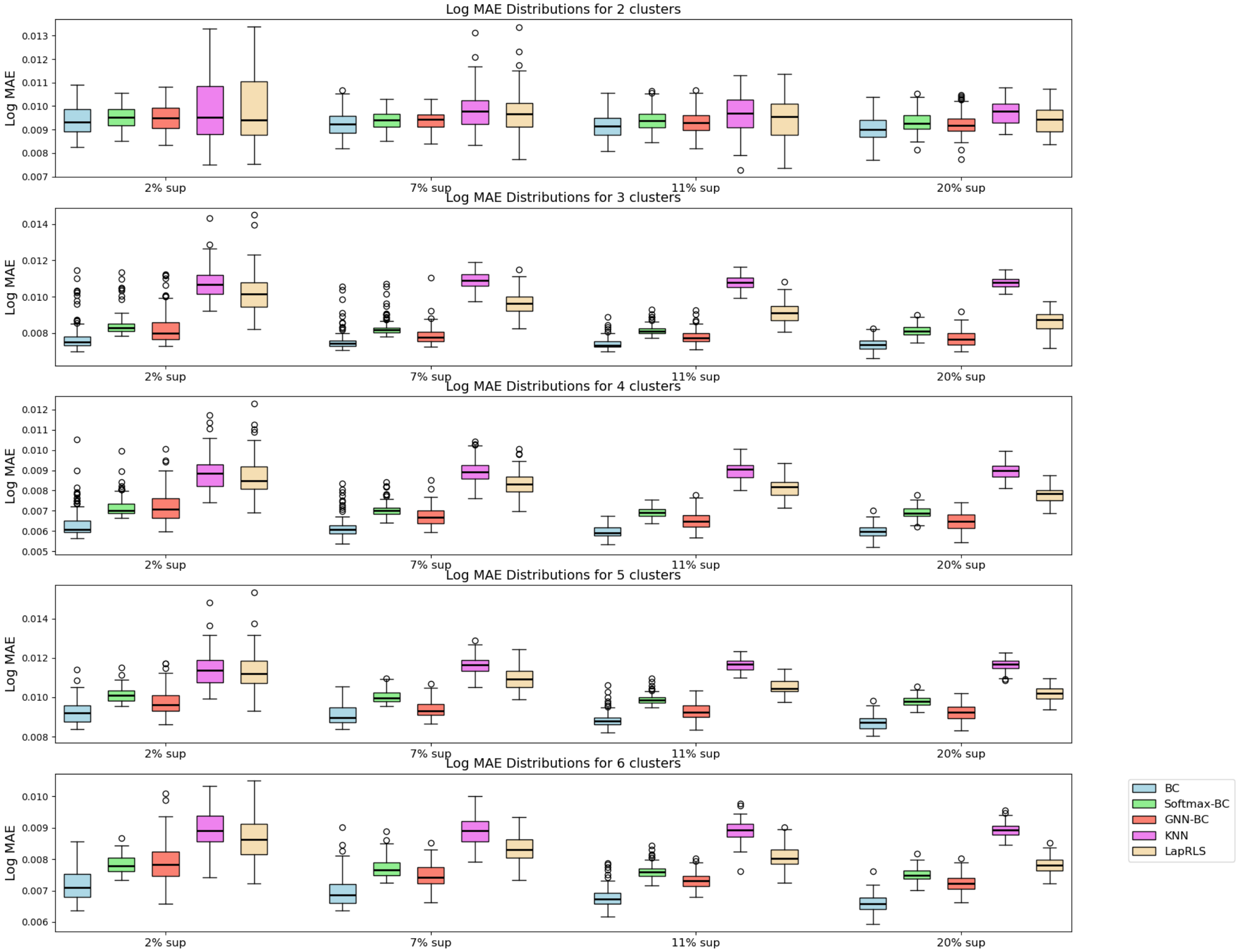

Figure 6. Log MAE Distribution Box Plots of Bridge Clustering (BC), Softmanx-BC, GNN-BC, KNN, and LapRLS over 2 - 6 clusters, and supervision of 2%, 7%, 11%, and 20%, corresponding to 1, 3, 5, and 10 supervised points. Circles outside the whiskers are outliers that are considered in quartile distributions.

[8] R. G. Nespolo, A. D. B. Valejo, and A. d. A. Lopes. A Study of Transductive Graph-Based Regression. *International Journal of Computer Information Systems and Industrial Management Applications*, 17:18–18, Jan. 2025. 2

[9] patrickpxye. food smart. https:https://github.com/patrickpxye/food_smart, note = GitHub. [Online; accessed Apr. 27, 2025], 2023. 2

[10] steubk. WikiArt dataset. https://www.kaggle.com/datasets/steubk/wikiart, 2024. Kaggle. [Online; accessed Apr. 27, 2025]. 2

[11] A. Tarvainen and H. 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. 2

[12] J. E. van Engelen and H. H. Hoos. A survey on semi-supervised learning. *Machine Learning*, 109(2):373–440, Feb. 2020. 1

[13] S. J. Wetzel, R. G. Melko, and I. Tamblyn. Twin neural network regression is a semi-supervised regression algorithm. *CoRR*, abs/2106.06124, 2021. 2

[14] P. Ye, Y. Wu, and E. Vitercik. Bridged clustering in scientific research. In *2nd AI4Research Workshop: Towards a Knowledge-grounded Scientific Research Lifecycle*. 3

[15] J. Zheng, L. Ye, and Z. Ge. Laplacian regularization of linear regression model for semi-supervised industrial soft sensor development. *Expert Systems with Applications*, 254:124459, Nov. 2024. 2

Generative AI to refine prose, in accordance with the Honor Code.

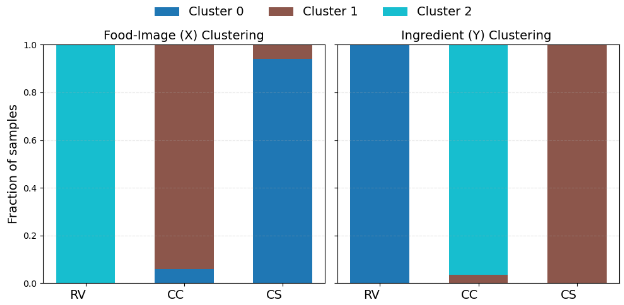

Figure 7. Clustering quality for 3 clusters, shown by cluster assignment consistency for different cuisines, in the food image feature space (X) and the ingredients output feature space(Y). RV is red velvet cake, CC is chicken curry, and CS is caprese salad.

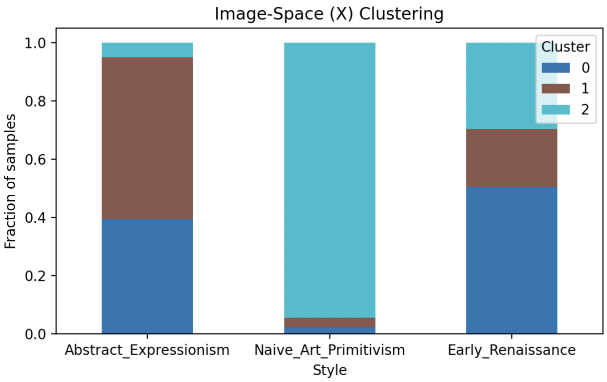

Figure 8. Clustering quality for 3 clusters for WikiArt.

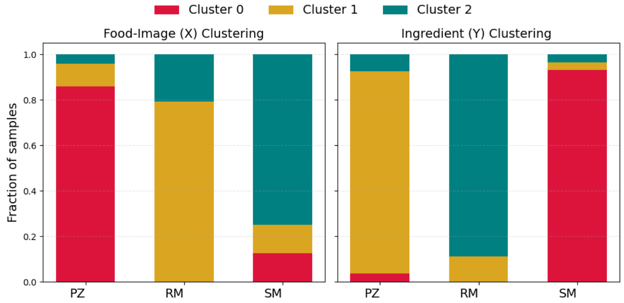

Figure 9. Clustering quality for 3 clusters, shown by cluster assignment consistency for different cuisines, in the food image feature space (X) and the ingredients output feature space(Y). PZ is pizza, RM is ramen, and SM is sashimi.