A Implementation details

We utilize ResNet-18 backbone [1] as $\phi_1(\cdot)$ pretrained on the train split of the corresponding dataset with MOCOv2 [2]. Features are obtained after *avgpool* with dimension equal to 512. For the ImageNet-1k dataset, we utilize ResNet-50 backbone as $\phi_1(\cdot)$ pretrained with MOCOv2. Here, features are obtained after *avgpool* with dimension equal to 2048. We use the same backbone and pretraining strategy for baselines as well. To enforce orthogonality constraint on the weights of the task encoder we apply Pytorch parametrizations [3]. When precomputing representations we employ standard data preprocessing pipeline of the corresponding model and do not utilize any augmentations during HUME's training. In all experiments, we use the following hyperapameters: number of iterations $T = 1000$, Adam optimizer [4] with step size $\alpha = 0.001$ and temperature of the sparsemax activation function $\gamma = 0.1$. We anneal temperature and step size by 10 after 100 and 200 iterations. We set regularization parameter η to value 10 in all experiments and we show ablation for this hyperparameter in Appendix B. To solve inner optimization problem we run gradient descent for 300 iterations with step size equal to 0.001. At each iteration we sample without replacement 10000 examples from the dataset to construct subset $(X_{tr}, X_{te}), |X_{tr}| = 9000, |X_{te}| = 1000$. Since STL-10 dataset has less overall number of samples, we use 5000 ($|X_{tr}| = 4500, |X_{te}| = 500$) in the inductive setting and 8000 ($|X_{tr}| = 7200, |X_{te}| = 800$) in the transductive setting. For the ImageNet-1000 dataset we use inner and outer step size equal to 0.1, number of inner steps equal to 100, sample 20000 examples with $|X_{tr}| = 14000, |X_{te}| = 6000$ on each iteration. We do not anneal temperature and step size for the ImageNet-1000 dataset and other hyperparameters remain the same. To reduce the gradient variance, we average the final optimization objective (Eq. 7) over 20 random subsets on each iteration. To stabilize training in early iterations, we clip gradient norm to 1 before updating task encoder's parameters. We use $N_{neigh} = 500$ in Algorithm G1 to construct reliable samples for semi-supervised learning.

B Robustness to a regularization parameter

HUME incorporates entropy regularization of the empirical label distribution in the final optimization objective (Eq. 7) to avoid trivial solutions, *i.e.*, assigning all samples to a single class. To investigate the effect of the corresponding hyperparameter η, we run HUME from 100 random initializations W_1 for each $\eta \in \{0, 1, 2, 5, 10\}$ on the CIFAR-10 dataset. Figure B1 shows results with different values of hyperaparameter η. The results show that η is indeed a necessary component of HUME objective, *i.e.*, setting $\eta = 0$ leads to degenerate labelings since assigning all samples to a single class is trivially invariant to any pair of representation spaces. Furthermore, the results show that HUME is robust to different positive values of the parameter η. We set $\eta = 10$ in all experiments.

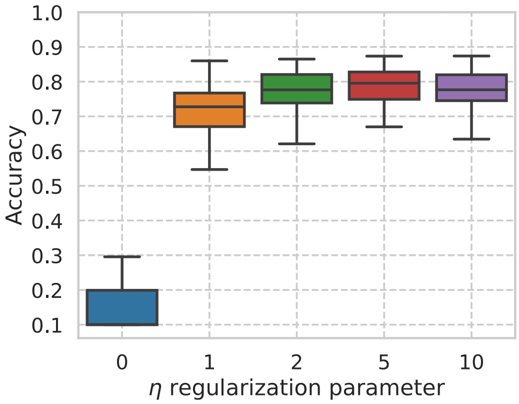


Figure B1: Performance of HUME on the CIFAR-10 dataset with different values of the entropy regularization. We use MOCOv2 self-supervised representations pretrained on the CIFAR-10 dataset and BiT large pretrained model.

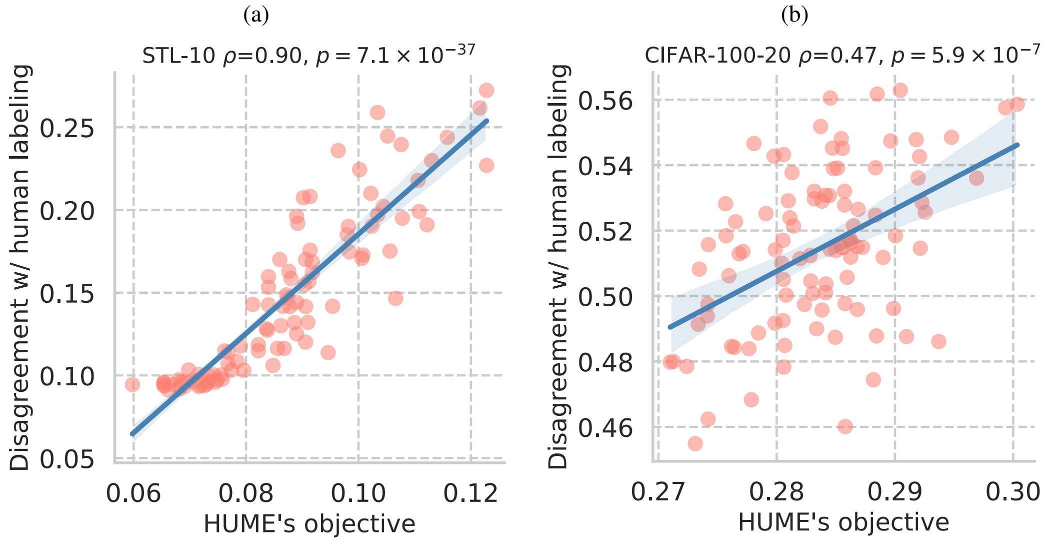
C Ablation study on different self-supervised methods

In all experiments, we use MOCOv2 [2] self-supervised representations. Here, we evaluate HUME's performance with different self-supervised learning method. In particular, we utilize SimCLR [5] and [6] to obtain representation space $\phi_1(\cdot)$. Table C1 shows the results in the inductive setting using DINO large pretrained model as the second representation space. The results show that HUME instantiated with MOCO consistently outperforms HUME instantiated with SimCLR on all of the datasets. This is expected result since MOCO representations are stronger also when assessed by a supervised linear probe. Alternatively, utilizing BYOL shows consistent improvements over MOCO representations. These results demonstrate that HUME can improve by employing stronger self-supervised representations. Interestingly, even with SimCLR representations HUME still outperforms unsupervised baselines pretrained with MOCOv2 in Table 2.

Table C1: Comparison of different self-supervised representations. We use DINOv2 large pretrained model. Stronger self-supervised representations lead to better performance.

	STL-10		CIFAR-10		CIFAR-100-20	
Method	**ACC**	**ARI**	**ACC**	**ARI**	**ACC**	**ARI**
SimCLR Linear	85.3	71.2	87.1	74.4	70.4	49.7
MOCO Linear	88.9	77.7	89.5	79.0	72.5	52.6
BYOL Linear	92.1	83.6	90.7	81.0	77.2	59.3
HUME SimCLR	86.9	74.3	85.2	71.8	51.8	33.9
HUME MOCO	90.8	81.2	88.4	77.6	55.5	37.7
HUME BYOL	91.5	82.7	89.8	79.7	56.0	40.3

D Correlation plots on the STL-10 and CIFAR-100-20 datasets.


(a)

(b)

Figure D1: Correlation plot between distance to ground truth human labeling and HUME's objective on the **(a)** STL-10 dataset and **(b)** CIFAR-100-20 dataset. HUME generates different labelings of the data to discover underlying human labeling. For each labeling (data point on the plot), HUME evaluates generalization error of linear classifiers in different representation spaces as its objective function. The value of ρ corresponds to Pearson correlation coefficient and p is the p-value of the corresponding two-sided test. HUME 's objective is well correlated with a distance to human labeling. In particular, tasks with lower HUME's objective tend to better match ground truth labeling, *i.e.*, $\rho = 0.9, p = 7.1 \times 10^{-37}$ on the STL10 dataset and $\rho = 0.47, p = 5.9 \times 10^{-7}$ on the CIFAR-100-20 dataset.

The HUME's objective is to search over the labelings of a dataset by minimizing a generalization error. To show the correlation between HUME's objective and the ground truth labeling, we plot correlation between generalization error of the labeling measured by cross-validation accuracy with respect to the found labeling and accuracy of the found labeling with respect to ground truth labeling. In addition to the results on the CIFAR-10 dataset presented in the main paper, Figure D1 shows the correlation plots on the STL-10 and CIFAR-100-20 datasets. The results demonstrate that HUME achieves the lowest generalization error for tasks that almost perfectly correspond to the ground truth labeling on the STL-10 dataset, allowing HUME to recover human labeling without external supervision. On the CIFAR-100-20 dataset even the supervised linear model on top of MOCOv2 self-supervised representations does not attain low generalization error (72.5% accuracy in Table 1). Consequently, HUME's performance also reduces, thus this additionally suggests that employing stronger representations will lead to better performance of HUME as also shown in Appendix C. Nevertheless, Figure D1b shows fairly-positive correlation ($\rho = 0.47, p = 5.9 \times 10^{-7}$) between distance to ground truth human labeling and HUME's objective, thus confirming the applicability of HUME to more challenging setups when one of the representation spaces might be insufficiently strong.

E Quality of the reliable samples produced by HUME

HUME can be used to produce reliable samples which can be further utilized with any semi-supervised learning (SSL) method. To measure the quality of reliable samples, we use two different statistics of the produced reliable samples: per class balance and per class accuracy. Per class balance measures number of samples for each ground truth class, *i.e.*, $\sum_{j \in R}[y_j = k]$, where R is the set of indices of produced reliable samples, y_j is the ground truth label of sample j, k represents one of the ground truth classes number and $[\cdot]$ corresponds to Iverson bracket. Per class accuracy measures the average per class accuracy of the corresponding set of the reliable samples with respect to ground truth labeling. We follow standard protocol for the evaluation of SSL learning methods [7] and consider 4, 100 samples per class on the STL-10 dataset and 1, 4, 25, 400 samples per class on the CIFAR-10 dataset. We provide results averaged across all classes in Table E1 and the corresponding standard deviations across classes in Table E2. In addition to the provided statistics, Figure E1 presents accuracies of the reliable samples on the STL-10 and CIFAR-100-20 datasets for wider range of number of reliable samples per class. Overall, the results show that on the STL-10 and CIFAR-10 datasets HUME shows almost perfect balance and mean per class accuracy, *i.e.*, up to 100 samples per class on STL-10 and up to 400 samples per class on CIFAR-10, thus demonstrating that HUME can produce reliable pseudo-labels for SSL methods.

Table E1: Mean per class balance and mean per class accuracy for the reliable samples produced by HUME on the STL-10, CIFAR-10 and CIFAR-100 datasets. Mean is computed over the number of classes in the corresponding dataset.

	STL-10			**CIFAR-10**				**CIFAR-100-20**		
Quantity	**4**	**100**	**1**	**4**	**25**	**400**	**1**	**10**	**50**	**100**
Mean Per Class Balance	4.0	100.0	1.0	4.0	25.0	400.0	0.6	6.3	26.3	52.6
Mean Per Class Accuracy	100.0	99.6	100.0	100.0	99.6	99.7	72.7	62.3	51.1	48.9

Table E2: Standard deviations of per class balance and per class accuracy for the reliable samples produced by HUME on the STL-10, CIFAR-10 and CIFAR-100 datasets. Standard deviations are computed over the number of classes in the corresponding dataset.

	STL-10			**CIFAR-10**				**CIFAR-100-20**		
Quantity	**4**	**100**	**1**	**4**	**25**	**400**	**1**	**10**	**50**	**100**
Mean Per Class Balance	0.0	1.4	0.0	0.0	0.5	1.3	0.6	5.2	23.6	45.6
Mean Per Class Accuracy	0.0	0.9	0.0	0.0	1.2	0.4	44.1	41.9	46.5	47.2

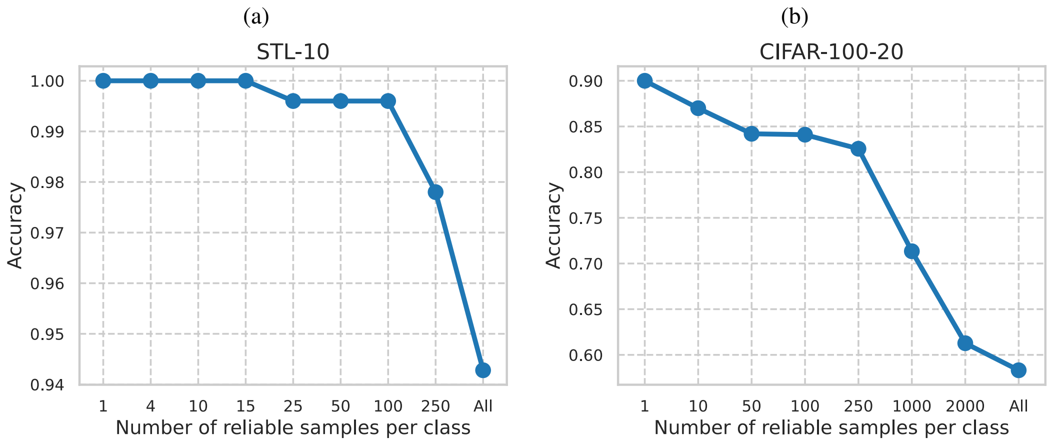


Figure E1: Accuracy of the reliable samples on the **(a)** STL-10 dataset and **(b)** CIFAR-100-20 dataset.

F Ablation study on different aggregation strategies on the STL-10 and CIFAR-100-20 datasets

We additionally study the effect of the proposed aggregation strategy on the STL-10 and CIFAR-100-20 datasets in an inductive setting. We employ MOCOv2 self-supervised representations as representation space $\phi_1(\cdot)$ and show the results for different large pretrained models as representation space $\phi_2(\cdot)$. Figure F1 shows the results for the STL-10 and CIFAR-100-20 datasets, respectively. We observe the similar behaviour to the results obtained in the main paper on the CIFAR-10 dataset. Namely, the proposed aggregation strategy stabilizes the results and provides robust predictions. It is worth noting that even using top-5 labelings in the majority vote is enough to produce stable results. For weaker models such as BiT, aggregation strategy has more effect on the performance and the optimal strategy is to aggregate around top-10 tasks. This is expected given the high correlation between HUME's objective and accuracy on ground truth labels since this strategy gives robust performance and tasks are closer to human labeled tasks. Larger models such as DINO show high robustness to the aggregation strategy. It is important to emphasize that in experiments we always report average across all tasks and do not optimize for different aggregation strategies.

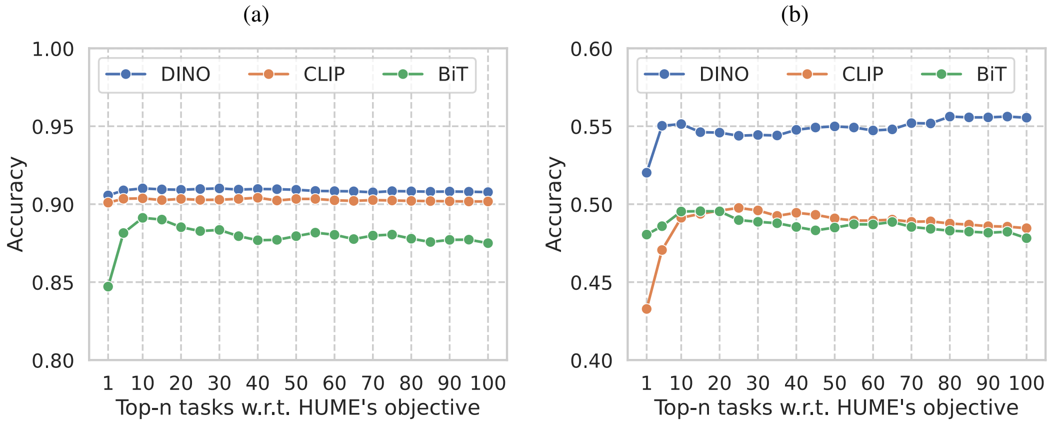


Figure F1: Different aggregation strategies on the **(a)** STL-10 dataset and **(b)** CIFAR-100-20 dataset. We use MOCOv2 self-supervised representations pretrained on the corresponding dataset and each line on the plot corresponds to the type of the large pretrained model.

G Algorithm for constructing reliable samples

We showed that HUME can be utilized to construct a set of reliable labeled examples to transform an initial unsupervised learning problem to a semi-supervised problem. It is worth noting that standard semi-supervised setting requires a balanced labeled set, *i.e.*, equal number of labeled samples for each class. For simplicity, we adapt the approach presented in SPICE [8] to produce a balanced set of reliable samples. Namely, we sort all samples per class by *(i)* number of labelings in the majority vote which predict the same class, and *(ii)* number of neighbours of the sample which have the same class. Thus, we consider a sample reliable if both quantities are high. Finally, given the sorted order we take the required number of samples to stand as a set of reliable samples. The proposed algorithm is outlined in Algorithm G1.

Algorithm G1 Reliable samples construction

```
Input: Dataset $\mathcal{D}$, number of classes $K$, number of samples per class $N_k$, number of neighbours
    $N_{neigh}$, self-supervised representation space $\phi_1(\cdot)$, trained labelings $\tau_1, \dots, \tau_m$
Compute majority vote: $\tau_{\text{MAJ}}(x) = \arg\max_{k=1,\dots,K} \sum_{i=1}^{m} \mathbb{1}[\tau_i(x) = k]$
Count number of agreed labelings:
$\mathcal{A}^{\tau}(x) = \sum_{i=1}^{m} \mathbb{1}[\tau_i(x) = \tau_{\text{MAJ}}(x)]$
Find nearest neighbours in representation space $\phi_1$:
$\mathcal{N}(x) \leftarrow N_{neigh}$ nearest neighbours for sample $x \in \mathcal{D}$
Count number of agreed nearest neighbours:
$\mathcal{A}^{\text{nn}}(x) = \sum_{z \in \mathcal{N}(x)} \mathbb{1}[\tau_{\text{MAJ}}(z) = \tau_{\text{MAJ}}(x)]$
Initialize set of reliable samples: $\mathcal{R} = \emptyset$
for $k = 1$ to $K$ do
    Take per class samples: $S_k = \{x \in \mathcal{D} | \tau_{\text{MAJ}}(x) = k\}$
    Sort $S_k$ in descending order by lexicographic comparison of tuples $(\mathcal{A}^{\text{nn}}(x), \mathcal{A}^{\tau}(x))$
    Take top-$N_k$ samples from the sorted $\hat{S}_k$ and update set of reliable samples:
    $\mathcal{R} = \mathcal{R} \cup \text{top-}N_k(\hat{S}_k)$
end for
```

References

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition*, 2016.

[2] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[3] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, et al. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, 2019.

[4] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2017.

[5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, 2020.

[6] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, et al. Bootstrap your own latent - a new approach to self-supervised learning. In *Advances in Neural Information Processing Systems*, 2020.

[7] Yidong Wang, Hao Chen, Qiang Heng, Wenxin Hou, Yue Fan, et al. FreeMatch: Self-adaptive thresholding for semi-supervised learning. In *International Conference on Learning Representations*, 2023.

[8] Chuang Niu, Hongming Shan, and Ge Wang. Spice: Semantic pseudo-labeling for image clustering. *IEEE Transactions on Image Processing*, 31:7264–7278, 2022.