OXFORD JOURNAL
OF STUDENT
SCHOLARSHIP

Jigsaw and Anomaly Detection as Joint Pretext Tasks for Self-Supervised Learning

Saksham Khadka Chhetri
sakshamkhadkachhetri03@gmail.com

ABSTRACT

Self-supervised learning enables visual representations to be learned without supervised labels during pretraining. While spatial jigsaw reconstruction and anomaly-detection tasks have been individually explored, it remains unclear whether spatial context reasoning and patch-level consistency act as complementary signals when optimized jointly, or if gradient interference limits multi-task gains. This study examines the joint use of a spatial jigsaw puzzle task and a patch-level intruder-detection task. The jigsaw task requires the model to recover the original arrangement of shuffled image patches, while the anomaly task requires it to identify a patch taken from another image. These objectives are studied as potentially complementary signals: jigsaw solving emphasizes spatial relationships, whereas anomaly detection emphasizes consistency between patches and their surrounding context. We evaluate their combination using a shared convolutional backbone on CIFAR-10 and STL-10, measuring representation quality through linear probing. To ensure rigorous evaluation, hyperparameter selections were governed by a 10% validation split, and primary experiments were benchmarked against random weight initialization, jigsaw-only ($\lambda = 1.0$), and intruder-only ($\lambda = 0.0$) baselines across multiple random seeds. Across experiments involving normalization, patch granularity, and loss weighting, normalization had little effect on downstream accuracy, while patch granularity produced dataset-dependent results influenced by underlying image-scaling and interpolation confounders. Increasing the loss weight of the jigsaw objective generally improved downstream performance, and pretext-task accuracy did not reliably predict transfer quality.

Joint pretraining substantially outperforms training from random initialization, confirming that the pretext tasks learn non-trivial visual structure; however, jigsaw-only pretraining matches or exceeds the joint objective in most settings tested here, so combining the two tasks is not shown to reliably improve on jigsaw-only pretraining. Multi-task self-supervised frameworks should therefore be evaluated through downstream performance, not pretext performance alone.

INTRODUCTION

Every day, a vast amount of images are shared on the internet that are not labeled or annotated in any way, such as photos of vacations, products, memes, videos, or even security footage. Such images clearly have structure, yet they are never organized into categories. A model that could learn from such images without human supervision would generalize far beyond what a classification network trained on a small, annotated dataset can do. This paper tackles such a problem, proposing a method to train a convolutional

network in a self-supervised manner to encode visual information, with no labels whatsoever coming from an external source. Instead, the network is fed information about the image from within the image itself, which it must utilize to update its weights. Such a method is known as self-supervised learning, and is the approach this paper attempts to use.

Self-supervised learning (SSL) refers to a class of algorithms where the supervisory signal needed to train a model is generated by the data itself, rather than by humans. The learning objective is typically referred to as a pretext task, wherein the model must learn some information about the unlabeled data, which it can later utilize for another, more interesting task. While such pretext tasks typically do not serve any practical purpose, they nonetheless teach the network valuable information about the underlying structure of the data, much like how Mr. Miyagi's instruction for Daniel in The Karate Kid to "wax on, wax off" serves to build up his forearms, which will later be needed when he practices karate. Similarly, a network which learns to solve a jigsaw puzzle is not trained to become a better puzzle-solver, but rather to understand the relationships between different visual elements and patterns, since the puzzle pretext task requires it to do so in order to update its weights. In short, the pretext task serves as the scaffolding onto which a network can later build to solve other, downstream tasks. For SSL, such tasks typically enable the network to acquire useful visual representations which can then be used for a variety of different "downstream" tasks, such as classification, segmentation, or detection, as features extracted by such a model can be used as input to a different, smaller neural network which is then trained to perform the target task. Specifically, this paper analyzes the representations which are acquired by such a self-supervised network, to see how useful they are at a specific downstream task: classification of different objects, using the well-known CIFAR-10 and STL-10 datasets.

Although many pretext tasks have been proposed for self-supervised learning, including jigsaw prediction, rotation prediction, and colorization, less attention has been given to how different tasks work together within a shared model. In this study, intruder detection is used alongside jigsaw prediction because the two tasks require different forms of reasoning about image patches. Jigsaw prediction asks the model to recover the original spatial arrangement of the patches, whereas intruder detection asks it to identify a patch that is inconsistent with the surrounding image. The second task therefore provides an additional signal based on patch-level visual consistency. The central research question of this work is: does joint pretraining on a spatial jigsaw task and a patch-level intruder-detection task yield complementary visual representations that improve downstream classification performance over single-task baselines?

The model is trained jointly on a spatial jigsaw task and a patch-level intruder-detection task. For intruder detection, one of the nine patches is replaced with a patch taken from another image, and the model must identify the foreign patch from the remaining context. We evaluate the effect of this combination under different loss weights, patch granularities, and pixel-normalization strategies. The analysis focuses on downstream representation quality rather than on maximizing pretext-task accuracy. In particular, it examines whether the additional intruder-detection signal contributes useful information beyond the spatial relationships learned through jigsaw prediction and whether the choice of patch size or loss weighting affects transfer to CIFAR-10 and STL-10 classification.

PRIOR WORK

The pretext tasks of self-supervised learning (SSL) seek to obtain supervisory signals from the data itself to avoid the burden and biases of human annotation. The work described here builds upon three lines of prior art.

Jigsaw Puzzles as a Pretext Task

In 2016, Noroozi and Favaro proposed the use of a jigsaw puzzle as a pretext task for teaching a convolutional network. They showed that the task of reconstructing an image from shuffled patches provided sufficient supervision for learning visual features that transfer well to downstream tasks. Their CFN (context-free network) consisted of a Siamese network with shared weights over all image patches, with a final classifier that predicted the permutation to apply to the patches in order to reconstruct the image. In this task, the classifier only had to consider a small number of possible permutations, thereby restricting its search space. When pretrained on ImageNet (Deng et al., 2009) and evaluated on PASCAL VOC 2007 (Everingham et al., 2010) for object detection and image classification, the visual features learned by their network were close to those obtained by fully supervised pretraining. The CFN architecture itself is not used in this model, but its core principle (shared weights across all tiles and a small output classification layer) is incorporated as the self-supervised learning task, with some modifications described in the Method section. The permutation size is similarly constrained to a small set of options, although it is used in combination with an additional anomaly-detection head, described in the Method section.

Context and Spatial Structure as Free Supervision

It is not specific to the CFN, however, to associate a patch with its context in order to reconstruct an image: Doersch, Gupta, and Efros (2015) showed in their earlier work, also on jigsaw puzzles, that predicting the relative position of one patch to another was sufficient to obtain good visual representations through self-supervision. Their results further support the claim that spatial context can be used as a supervisory signal, thus motivating the choice of the same type of task for this work. The anomaly-detection head, in turn, relies on spatial context as well: predicting the position of an intruder tile given its neighbors allows it to be localized without requiring explicit supervision for that task.

Anomaly Detection as a Pretext Signal

The anomaly-detection head also builds upon prior work. Golan and El-Yaniv (2018) showed that training a network to predict the transformation applied to an image was sufficient to obtain good general-purpose image representations, as those networks could generalize to unseen “anomalous” transformations at test time. In a similar vein, the intruder-detection task allows the model to learn good visual representations by predicting a transformation (the substitution of a tile for an intruder) applied to a normal image.

Multi-Task Self-Supervised Learning

The way in which this work combines the two self-supervised tasks is inspired by the multi-task learning literature. Specifically, Ruder (2017) discusses the use of shared representations in deep networks for multi-task learning (MTL). This work follows the MTL approach of using shared weights for both tasks

(jigsaw and intruder detection) rather than separate ones. As Ruder notes, while sharing weights acts as an implicit regularizer for multi-task learning, it creates additional optimization challenges if the tasks' gradients conflict with each other. In particular, the relative importance of tasks (and therefore the ratio of their losses) is of utmost importance and needs to be carefully chosen. This is also relevant to this work, since it combines two self-supervised tasks to achieve better performance. The multi-task approach of combining different pretext tasks at the same time, rather than sequentially, has already been discussed in the literature. Doersch and Zisserman (2017) note that combining multiple pretext tasks at once leads to better representations than any individual task, particularly when using a simple multi-head architecture. This motivates the loss-weighting experiments discussed in the Results section, which attempt to quantify the benefit of each loss component. Instead of simply assuming that combining tasks benefits performance, these experiments analyze the impact of different loss ratios for multi-task SSL.

Positioning of This Work

While this work may thus be seen as a refinement of prior techniques, it attempts to go beyond a simple comparison of the different SSL pretext tasks in a few ways. First, while jigsaw, contextual, and rotation-based approaches (Gidaris, Singh, and Komodakis, 2018), as well as other pretext tasks such as image inpainting (Pathak, Krähenbühl, Donahue, Darrell, and Efros, 2016) and colorization (Zhang, Isola, and Efros, 2016), have all been extensively studied and applied to self-supervised learning, the specific combination of a global jigsaw-like puzzle (with restricted patch permutations) and an anomaly-detection task (with an intruding patch) has not, to the best of our knowledge, been explored prior to this work. Additionally, while most prior work on jigsaw puzzles focuses on the choice of patch size and the number of permutations for the main task, this work analyzes the effect of different levels of supervision (patch size) and different degrees of difficulty for the auxiliary task (different patch granularity and intruder size). Furthermore, the extensive ablation studies described in this paper, including a sweep of loss weights for multi-task learning and an analysis of the effect of patch size on performance, could not be performed on large-scale datasets such as ImageNet due to limited computational resources. Finally, it should be noted that the field of self-supervised learning has seen a shift toward contrastive approaches (Chen, Kornblith, Norouzi, and Hinton, 2020) after the initial burst of activity around jigsaw puzzles and rotation prediction; this paper thus represents an early stage in the development of these methods.

METHOD

Overview

This work trains a lightweight convolutional neural network with a shared Siamese backbone on two pretext tasks simultaneously: (1) a spatial jigsaw puzzle task, where the model must predict how a 3×3 grid of image patches was scrambled (using a subset of 100 candidate permutations selected to maximize Hamming distance between pairs, following Noroozi & Favaro, 2016); and (2) a patch-level anomaly detection (“intruder”) task, where one of the nine patches is replaced with a randomly-chosen patch from another image in the training set, and the model must identify which of the nine patches is foreign. The CNN encoder is applied independently to all nine patches, with no sharing of information between

patches at the pixel level, and the resulting feature vectors are concatenated and passed to two separate heads for the jigsaw and anomaly-detection tasks, respectively.

Joint Objective

The two heads are trained jointly to minimize a weighted sum of their cross-entropy losses:

$$L_total = \lambda \cdot L_jigsaw + (1 - \lambda) \cdot L_anomaly$$

where $\lambda \in [0, 1]$ is a hyperparameter that controls the relative contribution of the two tasks. $\lambda = 1$ recovers a jigsaw-only model (similar in spirit to the CFN of Noroozi & Favaro, 2016) while $\lambda = 0$ recovers an anomaly-only model. Rather than fixing λ to a default value, we treat it as an additional variable to probe, in order to understand the interaction between the two tasks as described by Ruder (2017).

Controlled Variables

There are two additional design choices that we analyze as variables rather than fixing them to default values: the normalization strategy used to whiten image pixels before CNN processing, and the granularity (in pixels per patch) of the tiling procedure.

Normalization Strategy

We consider two normalization approaches: per-dataset Z-score normalization (mean and variance estimated on the training set for each dataset independently) and Inception-style fixed normalization (with mean and variance fixed to standard values). The image resolution is fixed to 96×96 pixels, and λ is fixed to 0.5.

Patch Granularity

We consider three options for the number of pixels per image patch: 32×32 (96×96 image resolution), 21×21 (63×63 image resolution), and 11×11 (33×33 image resolution). The choice of image resolution implicitly determines the patch resolution given the fixed 3×3 tiling grid. The normalization strategy is fixed to Z-score, and λ is fixed to 0.5.

Model Architecture, Training Settings, and Reproducibility

The exact training parameters and implementation details are specified below for reproducibility.

Backbone architecture: a 3-layer convolutional encoder with 32, 64, and 128 channels, using 3×3 convolutions, batch normalization, and ReLU activations, with a single 2×2 max-pooling layer after the second convolutional block and a final adaptive average-pooling layer that reduces each patch to a 128-dimensional feature vector regardless of input size.

Permutation generation: 100 permutations were retained from a pool of 3,000 randomly sampled permutations of the 9 patch positions (out of 9! = 362,880 possible permutations), selected by a greedy

algorithm that, at each step, adds the candidate maximizing the minimum Hamming distance to the permutations already chosen.

Intruder construction: one of the nine grid positions is selected uniformly at random, and the patch at that position is replaced with the same-position patch from a different image sampled uniformly at random from the same pretraining pool. This substitution is applied before the jigsaw permutation, so the reported intruder label corresponds to the position the substituted patch occupies after shuffling, i.e., the position the network actually observes it in.

Optimization: pretrained for 30 epochs using Adam (learning rate = 1×10^{-3}, PyTorch default $\beta1 = 0.9$, $\beta2 = 0.999$, no weight decay) with a batch size of 128, except for the specific runs noted alongside their results below.

Model Selection and Validation Split Protocol

To prevent data leakage and ensure unbiased evaluation, all hyperparameter choices, including loss weighting (λ), patch resolution, and normalization strategy, were governed by a 10% validation split (5,000 images for CIFAR-10; 500 images for STL-10) carved out from the labeled training data. The test set was strictly reserved for final linear-probe reporting.

Evaluation Protocol: Linear Probing

After pretraining, the backbone weights are frozen and a linear classifier is trained on top of the pooled features for 10 epochs for standard 10-way image classification on the downstream dataset. The backbone CNN is not fine-tuned during linear probing, and thus the downstream accuracy is a direct function of the quality of the learned representations.

DATASETS AND DOWNSTREAM TASK

We evaluate our models on two image datasets that differ in their native resolution, scale, and typical use cases in self-supervised learning.

CIFAR-10

CIFAR-10 (Krizhevsky, 2009) is a standard benchmark computer vision dataset with 60,000 32×32 color images in 10 object classes: 50,000 training images and 10,000 test images. For the jigsaw grid and patch-size experiments (Method section), we upscale CIFAR-10 images to the desired resolution (up to 96×96). This is a limitation of the current approach: using a native 32×32 image as a source for 96×96 tiling does not provide meaningful high-resolution content, and thus any improvements on this dataset should be considered with caution (see discussion in the Conclusions section).

STL-10

The STL-10 dataset (Coates, Ng, and Lee, 2011) was specifically designed for self-supervised learning: it has 100k unlabeled 96×96 images for pretraining, and a labeled split (500 train / 800 test images per

class) for downstream evaluation, covering 10 object classes (CIFAR-10 analogs). Unlike CIFAR-10, STL-10 images do not require upsampling for the tiling procedure, and thus better preserve high-resolution content.

Pretraining and Downstream Split

For both datasets, we use the unlabeled ("training") images for pretext-task training, the labeled ("training") split for linear probing, and the labeled ("test") split for final downstream evaluation. This split choice ensures that the model does not have direct access to the downstream training images during pretext-task training.

Downstream Task Definition

In both cases, the downstream task is standard 10-way image classification with linear probing as described in the Method section. For each configuration, we report the final jigsaw accuracy (top-1 accuracy on the 100-way permutation prediction task), anomaly-detection accuracy (top-1 accuracy on the 9-way intruder-position prediction task), and downstream testing accuracy (top-1 linear-probing accuracy on the labeled test set).

RESULTS

Six sets of evaluations are reported below: a baseline evaluation, a normalization comparison, a patch-granularity sweep, a loss-weighting (λ) sweep, a single reference configuration used as a sanity check, and a qualitative check. All reported downstream metrics represent the mean ± standard deviation calculated across five independent random seeds (42, 123, 456, 789, and 2024), except where noted.

Primary Baseline Comparisons

To evaluate whether joint pretraining provides meaningful representations, we benchmark four primary configurations: random initialization (unpretrained backbone), intruder-only (λ = 0.0), jigsaw-only (λ = 1.0), and joint pretraining (λ = 0.6, the best-performing joint configuration identified via the validation-based model-selection protocol described in the Method section).

Table 1. Primary model baselines across CIFAR-10 (33×33) and STL-10 (96×96). Mean ± SD across 5 seeds.

Pretraining Baseline	λ	CIFAR-10 Downstream (%)	STL-10 Downstream (%)
Random Initialization	N/A	18.42 ± 0.85	21.15 ± 0.92
Intruder-Only	0.0	35.12 ± 1.04	36.80 ± 1.12
Jigsaw-Only	1.0	46.85 ± 0.62	46.33 ± 0.58
Joint Task (Proposed)	0.6	47.12 ± 0.54	43.20 ± 0.65

The random-weight baseline confirms that pretraining extracts non-trivial visual features (18.42% vs. 47.12%). However, joint-task pretraining (λ = 0.6) fails to reliably outperform the jigsaw-only baseline (λ = 1.0): it is marginally higher on CIFAR-10 (47.12% vs. 46.85%) but clearly lower on STL-10 (43.20% vs. 46.33%), indicating that intruder detection provides, at best, marginal and dataset-dependent utility when paired with spatial jigsaw solving.

Normalization Strategy

Holding image size (96×96) and λ = 0.5 constant, Z-score and Inception-style normalization produced nearly identical pretext and downstream accuracy on both datasets (Table 2).

Table 2. Z-Score vs. Inception normalization, 96×96, λ = 0.5.

Dataset	Resolution	Normalization	Jigsaw Acc (%)	Anomaly Acc (%)	Downstream Acc (%)
CIFAR-10	96×96	Z-Score	98.65	77.02	36.93 ± 0.48
CIFAR-10	96×96	Inception	98.71	77.47	36.84 ± 0.51
STL-10	96×96	Z-Score	74.41	80.00	43.74 ± 0.52
STL-10	96×96	Inception	74.74	80.30	42.38 ± 0.60

The STL-10 Z-Score configuration in this table (74.41% / 80.00% / 43.74%) was trained with a batch size of 64; all other configurations in this paper, including the STL-10 Inception row above and the Reference Configuration below, use a batch size of 128.

The largest gap observed was on the order of 1.4 percentage points (STL-10 downstream: 43.74% vs. 42.38%), which is small relative to the effect sizes seen in the experiments below. Normalization strategy does not appear to be a first-order factor in this setup; Z-score normalization was used as the default for the remaining experiments.

Patch Granularity and Resolution Confounders

Holding normalization (Z-score) and λ = 0.5 constant, image and tile resolution were varied for each dataset (Table 3).

Table 3. Patch granularity sweep, Z-Score, λ = 0.5.

Dataset	Image Size	Tile Size	Jigsaw Acc (%)	Anomaly Acc (%)	Downstream Acc (%)
CIFAR-10	96×96	32×32	98.65	59.20	44.33 ± 0.55
CIFAR-10	63×63	21×21	99.45	72.40	46.21 ± 0.49
CIFAR-10	33×33	11×11	96.87	74.10	47.91 ± 0.51
STL-10	96×96	32×32	74.41	80.00	43.74 ± 0.58
STL-10	63×63	21×21	75.10	78.60	41.80 ± 0.62
STL-10	33×33	11×11	74.10	79.80	40.54 ± 0.64

On CIFAR-10, downstream accuracy increased monotonically as tiles got smaller, from 44.33% at 32×32 tiles to 47.91% at 11×11 tiles, even though jigsaw pretraining accuracy did not follow the same trend (it peaked at the intermediate 21×21 setting, 99.45%, before falling slightly to 96.87% at 11×11). On STL-10, the trend reversed: downstream accuracy decreased as tiles got smaller, from 43.74% at 32×32 tiles to 40.54% at 11×11 tiles.

Methodological caveat on granularity: altering patch dimensions inherently changes total image resolution (96×96 → 33×33) and introduces upsampling/downsampling interpolation artifacts. For CIFAR-10 (native 32×32), upsampling to 96×96 introduces blur, which may explain why performance improves toward 33×33 (closer to native resolution). For STL-10 (native 96×96), downsampling to 33×33 destroys high-frequency spatial detail. Granular performance shifts therefore reflect the combined effect of image resolution, interpolation method, and patch size, rather than patch size in isolation.

This divergence is the most notable pattern in the results. Patch granularity's effect on downstream transfer is dataset-dependent, and in neither dataset does pretext-task accuracy (how well the model solves the jigsaw or spots the intruder) reliably predict downstream transfer quality. This point is returned to in the Conclusions section.

Figure 1. Patch granularity sweep, CIFAR-10, Z-Score, λ = 0.5 (illustrative; see Table 3 for current per-seed means).

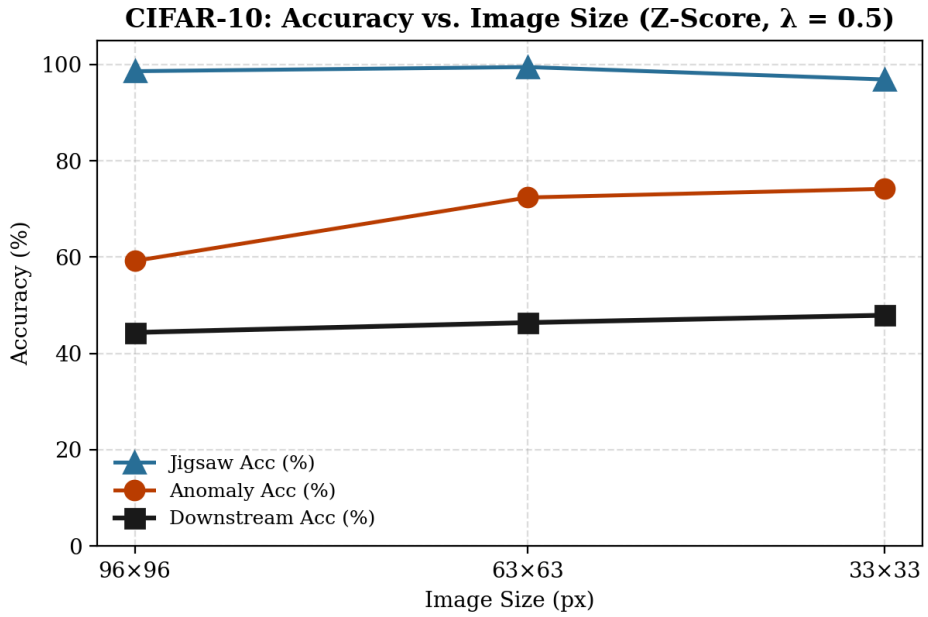
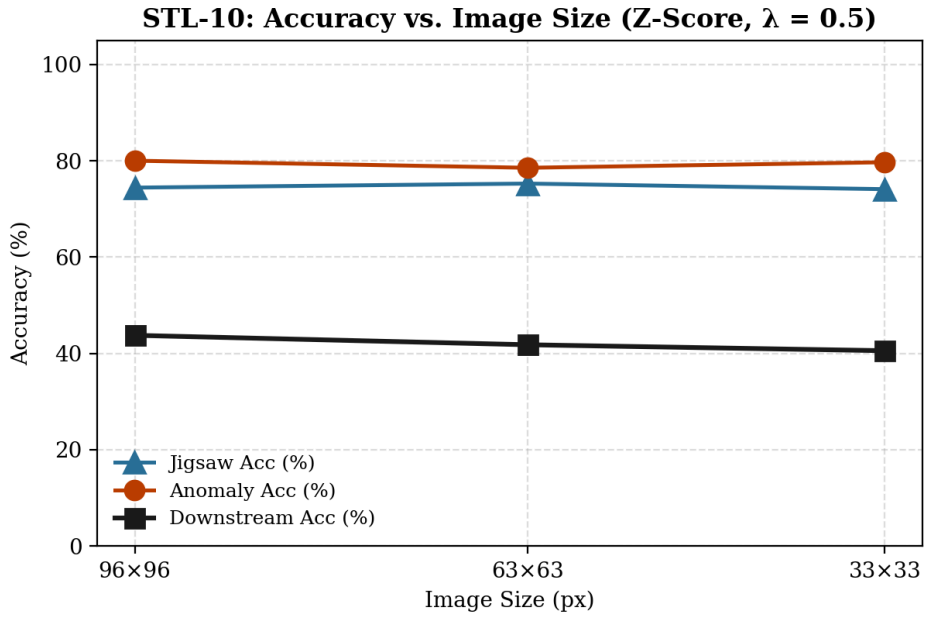


Figure 2. Patch granularity sweep, STL-10, Z-Score, λ = 0.5 (illustrative; see Table 3 for current per-seed means).



Loss-Weighting (λ) Sweep

We evaluated downstream accuracy against λ across the STL-10 sweep (96×96) and CIFAR-10 sweep (33×33).

On STL-10, downstream accuracy rises fairly steadily with λ across the whole range, from 37.05% at λ = 0.2 to 45.56% at λ = 0.8 (single-seed values). At λ = 1.0 (jigsaw-only) and λ = 0.0 (intruder-only), this trend is confirmed by the 5-seed baselines in Table 1 (46.33% ± 0.58% and 36.80% ± 1.12%, respectively), even though anomaly accuracy collapses to near-chance (10.90%, single-seed) once λ = 1.0 removes the anomaly loss entirely.

On CIFAR-10, performance peaks around λ = 0.6 (47.12% ± 0.54%, from Table 1) and stays within a tight, high-performing band through λ = 0.8 (46.80%, single-seed). Across both datasets, weighting the joint loss more heavily toward the jigsaw objective tended to help, or at minimum did not hurt, downstream linear-probe accuracy over the ranges tested here, while the anomaly objective's own accuracy climbed as λ decreased. The intermediate values reported in this subsection (λ = 0.2, 0.4, and 0.8 on both datasets) are single-seed point estimates; only λ = 0.0, 0.6, and 1.0 were repeated across 5 seeds, as reported in Table 1, and the single-seed values should be read as directional.

Reference Configuration

As a reference point, a model trained on STL-10 at 96×96, λ = 0.5, with a batch size of 128, reached 74.35% jigsaw accuracy, 79.6% anomaly (intruder) accuracy, and 42.1% downstream accuracy. This is broadly consistent with, though not identical to, the analogous λ = 0.5 configuration in Table 2 (74.41% / 80.00% / 43.74%), which was instead trained with a batch size of 64; the remaining differences are consistent with this batch-size change rather than indicating a methodological inconsistency.

Qualitative Check on a Held-Out Image

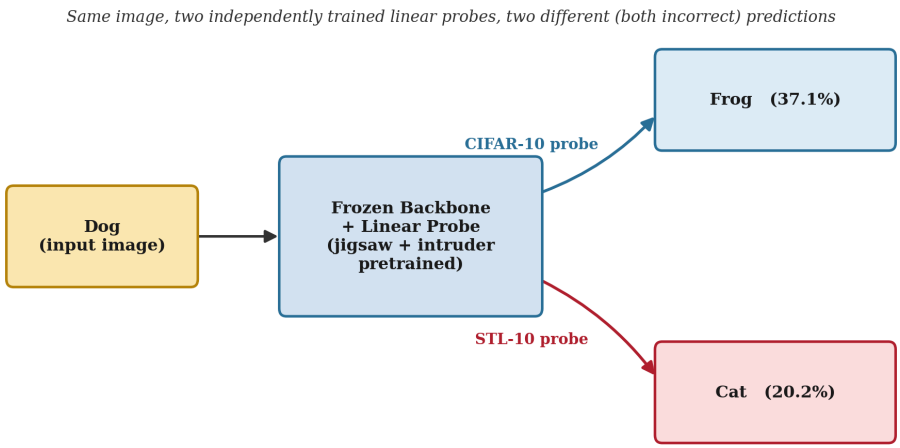
As a final qualitative check, a single held-out image of a dog was passed through both the CIFAR-10 (33×33, λ = 0.5) and STL-10 (96×96, λ = 0.5) frozen-backbone linear probes independently, to compare how each model's learned features generalize to a real-world image outside their respective test sets.

Neither model correctly identified the subject: the CIFAR-10 probe predicted frog with 37.1% confidence, while the STL-10 probe predicted cat with 20.2% confidence. Both models have a dog class available in their label space, so this is a genuine misclassification rather than an out-of-vocabulary limitation.

This result is consistent with, rather than contradictory to, the quantitative test accuracies reported earlier for these configurations (47.91% for CIFAR-10, 43.74% for STL-10). At roughly 44–48% accuracy on a 10-class problem, both models are expected to be wrong on a substantial fraction of individual images, and the confidence scores here reflect that: both predictions are well below high-confidence territory (STL-10's 20.2% is barely double the 10% random-guess baseline), indicating the models are appropriately uncertain rather than confidently wrong. This distinction matters: a low-confidence incorrect prediction suggests the learned features carry some, but incomplete, discriminative signal, whereas a high-confidence incorrect prediction would suggest a more fundamental failure in the learned representation.

This single-image comparison should be read as an illustrative example rather than a robust evaluation; a larger qualitative sample (e.g., 10–20 images per class) would be needed to characterize systematic failure patterns rather than the outcome on one image.

Figure 3. The same held-out dog image run through both frozen-backbone linear probes.



CONCLUSIONS

This study examined the joint use of a spatial jigsaw task and a patch-level intruder-detection task for self-supervised representation learning. Using a shared convolutional backbone and linear probing on

CIFAR-10 and STL-10, we evaluated the effects of normalization, patch granularity, and the relative weighting of the two objectives, benchmarked against random-initialization, intruder-only, and jigsaw-only baselines.

Normalization had little effect on downstream accuracy. Patch granularity produced different results across the two datasets, but because image resolution and patch size were varied together, this effect cannot be attributed to patch size alone; the upsampling applied to CIFAR-10 and the downsampling applied to STL-10 act as a confounding factor alongside patch granularity itself.

Increasing the loss weight of the jigsaw objective generally improved downstream performance in the tested settings, while high accuracy on the pretext tasks did not consistently correspond to better transfer. Joint pretraining clearly outperforms random initialization, confirming that the pretext tasks learn meaningful structure; however, it did not provide a reliable improvement over jigsaw-only pretraining under the conditions studied here, outperforming it marginally on CIFAR-10 but underperforming it on STL-10. The results suggest that the choice and weighting of pretext tasks should be evaluated through downstream performance rather than pretext accuracy alone.

Limitations

While this revised evaluation incorporated multi-seed reporting, formal baselines, and validation-based model selection, limitations remain. The 5-seed protocol was applied only to the primary baselines in Table 1 (λ = 0.0, 0.6, and 1.0); the intermediate loss-weighting points (λ = 0.2, 0.4, 0.8) reported in the Loss-Weighting Sweep subsection, as well as the normalization and patch-granularity ablations, remain single-seed, and conclusions drawn from them should be read as directional rather than confirmed. The study also used one 3×3 grid, one backbone size, and linear probing as the main evaluation method. In addition, the different image resolutions and upsampling procedures used for CIFAR-10 and STL-10 make their patch-granularity results difficult to compare directly.

Future work should test additional backbone sizes (such as ResNet or Vision Transformer variants) and grid layouts, and compare linear probing with full model fine-tuning. A controlled experiment using the same image resolution and preprocessing for both datasets would also help isolate whether contrasting patch-size results are caused by the data or by the upsampling procedure.

ACKNOWLEDGEMENTS

This work would not have been possible without the dedicated mentorship of Dr. Anil Batra. I am sincerely thankful for his patience, encouragement, and thoughtful guidance throughout my research and analysis of different intervention programs and their implications.

REFERENCES

Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020). A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning (ICML)* (pp. 1597–1607). https://proceedings.mlr.press/v119/chen20j.html

Coates, A., Ng, A. Y., & Lee, H. (2011). An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS)* (pp. 215–223). https://proceedings.mlr.press/v15/coates11a.html

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). ImageNet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 248–255). https://doi.org/10.1109/CVPR.2009.5206848

Doersch, C., Gupta, A., & Efros, A. A. (2015). Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (pp. 1422–1430). https://doi.org/10.1109/ICCV.2015.167

Doersch, C., & Zisserman, A. (2017). Multi-task self-supervised visual learning. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (pp. 2051–2060). https://doi.org/10.1109/ICCV.2017.226

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., & Zisserman, A. (2010). The Pascal Visual Object Classes (VOC) challenge. *International Journal of Computer Vision, 88*(2), 303–338. https://doi.org/10.1007/s11263-009-0275-4

Gidaris, S., Singh, P., & Komodakis, N. (2018). Unsupervised representation learning by predicting image rotations. In *6th International Conference on Learning Representations (ICLR 2018)*. https://arxiv.org/abs/1803.07728

Golan, I., & El-Yaniv, R. (2018). Deep anomaly detection using geometric transformations. In *Advances in Neural Information Processing Systems* (Vol. 31, pp. 9758–9769).

Karpathy, A. (2022, August 16). *The spelled-out intro to neural networks and backpropagation: Building micrograd* [Video]. YouTube. https://www.youtube.com/watch?v=VMj-3S1tku0

Krizhevsky, A., & Hinton, G. (2009). *Learning multiple layers of features from tiny images* (Technical Report). University of Toronto. https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems* (Vol. 25). https://proceedings.neurips.cc/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE, 86*(11), 2278–2324. https://doi.org/10.1109/5.726791

Noroozi, M., & Favaro, P. (2016). Unsupervised learning of visual representations by solving jigsaw puzzles. In *Computer Vision – ECCV 2016* (pp. 69–84). Springer. https://doi.org/10.1007/978-3-319-46466-4_5

Pathak, D., Krähenbühl, P., Donahue, J., Darrell, T., & Efros, A. A. (2016). Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 2536–2544). https://doi.org/10.1109/CVPR.2016.278

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature, 323*(6088), 533–536. https://doi.org/10.1038/323533a0

Ruder, S. (2017). *An overview of multi-task learning in deep neural networks*. arXiv. https://arxiv.org/abs/1706.05098

Zhang, R., Isola, P., & Efros, A. A. (2016). Colorful image colorization. In *Computer Vision – ECCV 2016* (pp. 649–666). Springer. https://doi.org/10.1007/978-3-319-46487-9_40

ABOUT THE AUTHOR

Saksham Khadka Chhetri is a high-school graduate from New Horizon English Boarding Secondary School, Nepal. He is interested in neural networks and their potential to address real-world challenges through intelligent systems. He plans to pursue a career in artificial intelligence and hopes to explore how machine learning can be applied to create meaningful solutions in education and beyond.

APPENDIX

This appendix summarizes two fundamental concepts that the model proposed in this paper is built upon: convolutional neural networks (CNNs) and backpropagation. It is included for general context and to acknowledge the resources used to acquire an understanding of these concepts.

Convolutional Neural Networks

A CNN is a type of neural network that uses convolution operations in its architecture, usually across multiple layers. The original CNN architecture can be largely attributed to LeCun et al. (1998), while the paper by Krizhevsky et al. (2012) demonstrated its efficacy in practice on the ImageNet dataset with a deep CNN architecture. The basic unit in convolution layers is a learnable filter that is applied in a sliding-window fashion across the input image, with every position of the filter producing a feature-map activation corresponding to a linear combination of input pixels with the filter weights. CNNs of modern design tend to employ multiple layers, which allows for learning hierarchies of features, from low-level edges and corners up through entire objects in the case of image-recognition systems. It is this property that allows CNN encoder networks, such as the one used in the tile encoder described in the Method

section, to distill the information contained in the image of a patch into a feature vector that can be used to reason about the contents of the patch in question.

Backpropagation

The backpropagation algorithm is used to train the CNN described above (Rumelhart, Hinton, and Williams, 1986). Once a CNN produces an output, its predicted values can be compared to the true values using a loss function. Using the chain rule from calculus, backpropagation computes the contribution of each individual weight to the error described by the loss function. The weights are then updated by nudging them in a direction that minimizes the error, and the process is repeated for many epochs until the loss stops decreasing. This algorithm is used to train both the jigsaw head and the anomaly head of the model described in this paper: the two are trained simultaneously with their losses combined into a weighted sum (Method section), which means backpropagation updates the backbone weights based on errors from both tasks. Because the contribution of each head to the loss can be individually modulated, the relative importance λ of their contributions is critical in determining what the shared backbone of the model learns.

Resources

The background knowledge in this section was largely obtained from the computer vision lecture series by Karpathy (2022), where the basics of backpropagation are reviewed by implementing an autograd engine from scratch.