Investigating the performance of deep learning methods for Hi-C resolution improvement

Ghulam Murtaza[1],
Atishay Jain[1], Madeline Hughes[1], Thulasi Varatharajan[2], and Ritambhara Singh[1,3,*]

[1]*Department of Computer Science, Brown University*
[3]*Center for Computational Molecular Biology, Brown University*
[2]*Department of Biology, Dietrich School of Arts and Sciences at University of Pittsburgh*
[*]*Corresponding Author*

Abstract

Motivation: HiC is a widely used technique to study the 3D organization of the genome. Due to its high sequencing cost, most of the generated datasets are of coarse quality, consequently limiting the quality of the downstream analyses. Recently, multiple deep learning-based methods have been proposed to improve the quality of these data sets by increasing their resolution through upscaling. However, the existing works do not thoroughly evaluate these methods using HiC reproducibility metrics across different HiC experiments to establish their applicability in real-world scenarios. This study extensively compares deep learning-based HiC upscaling methods on real-world, low-quality HiC datasets. We evaluate these models using HiC reproducibility metrics on data from three cell lines – GM12878 (lymphoblastoid), K562 (human erythroleukemic), and IMR90 (lung fibroblast) – obtained from different HiC experiments.
Results: We show that the deep-learning techniques evaluated in this study, trained on downsampled data, cannot upscale real-world, low-quality HiC matrices effectively. More importantly, we also show that retraining these methods on examples of real-world data improves their performance and similarity on target experimental data sets. However, even with retraining, our downstream analyses on the output of these methods suggest that these methods fail to capture the biological signals in the real-world inputs with high sparsity. Therefore, our study highlights the need for rigorous evaluation and identifies specific areas that need improvement concerning current deep learning-based HiC upscaling methods.
Availability: Implementation of our evaluation pipeline is available at `https://github.com/rsinghlab/Investigation-of-HiC-Resolution-Improvement-Methods`

1 Introduction

3D organization of the genome plays a vital role in cell fate and disease onset. A high-throughput chromosome conformation capture experiment, or HiC, is a genome-wide sequencing technique that allows researchers to understand and study the 3D organization of the genome [8]. Each read pair, sequenced using HiC, corresponds to an observed 3D contact between two genomic loci. This contact information captures both local and global interactions of the genomic regions. In the past decade, analysis of HiC data has facilitated many studies that have led to the discovery of important genomic structural features, including but not limited to A/B compartments [8] that denote active and inactive genomic regions, topologically associated domains (TADs) [1] that represent highly interactive genomic regions, and enhancer-promoter interactions [12] that are involved in the regulation of genes. Therefore, HiC experiments have proven to be crucial in advancing our understanding of the spatial structure of the genome and its relationship with gene regulation machinery.

When studying the spatial structure of the DNA, the quality of the downstream analysis of a HiC experiment is highly dependent on the quality of its contact map. Data from a HiC experiment coalesces into a matrix (or contact map) in which rows and columns correspond to fixed-width windows ("bins") tiled along the genomic axis, and values in the matrix are counts of read pairs that fall into the corresponding bins. The bin size typically ranges from 1 kilo basepair (kbp) to 1 mega basepair (mbp), where the choice of the bin size depends on the number of paired end-reads or the quality of the experiment. Low-quality experiments result in sparser read counts that require large bin sizes to account for the sparsity, resulting in a "low-resolution" contact map. Consequently, the downstream analysis of the generated matrix does not yield fine genomic features like enhancer-promoter interactions that typically occur in the 5 kbp to 10 kbp range [5]. Similarly, the output of a high-quality HiC experiment with a high number of read counts results in a "high-resolution" contact map with small bin sizes. This map enables the identification of the fine-grained genomic features. However, due to the quadratic scaling of the sequencing cost, most of the HiC datasets have relatively low read counts and, consequently, low-resolution ($\geq$ 40 kb) contact maps [14]. Constructing HiC matrices with high resolution, possibly requiring billions of reads [5], is often infeasible. The absence of such matrices makes the comprehensive analysis of the spatial structure of the DNA difficult.

Recently, researchers have proposed several computational methods to upscale [1] the HiC matrices by imputing the missing contacts. Notably, deep-learning-based methods have had remarkable success in upscaling HiC matrices. HicPlus [14], and HiCNN [10] employed convolutional layers with mean squared error loss to optimize the model weights to generate an upscaled HiC matrix conditional on downsampled low-quality input matrices. HiCGAN [9], DeepHiC [4], and VeHiCle [3] used GANs (Generative Adversarial Networks) to optimize the discriminator loss, guiding the model to produce HiC matrices with sharper and more realistic features in comparison to HiCPlus and HiCNN. These methods (except VeHiCle) report improved performance on synthetically downsampled HiC datasets using image-based evaluation metrics like correlation. However, neither synthetic datasets nor correlation metrics capture the underlying properties of the real-world HiC datasets because they make broad simplifying assumptions about the real-world data. Therefore, these results cannot provide insights into the applicability and performance benefits of using these deep learning methods for the real-world HiC upscaling task. Furthermore, while VeHiCle does utilize real-world datasets for training and evaluation, it restricts its analysis to four samples (across four cell lines) originating from the same experiment [3]. As a result, it does not assess the performance benefits on the datasets with varying levels of sparsity originating from different experiments. Consequently, it is unclear if these deep-learning-based methods trained on a subset of HiC datasets can generalize to datasets potentially belonging to different cell lines, sparsity-levels, and experiments. Therefore, it is critical to thoroughly evaluate these techniques using HiC specific similarity metrics across datasets collected from various sources to unveil the underlying trade-offs of using them in real-world scenarios.

In this study, we design an extensive set of experiments that compare the performance of deep learning-based HiC upscaling methods on real-world low-quality HiC datasets. We evaluate these models on six HiC contact maps, spanning three cell lines - GM12878 (lymphoblastoid), K562 (human erythroleukemic), and IMR90 (lung fibroblast) - obtained from different HiC experiments. First, we compare the upscaling performance of these methods using HiC-specific reproducibility (or similarity) scores. We find that the performance of all deep learning models deteriorates substantially compared to their performance on synthetically downsampled HiC map inputs. Our results show that the performance of HiCNN (the best generalizing model) degrades by almost 50% for one of the similarity score metrics. To alleviate this disparity, we retrain the best performing deep learning model with different input strategies, such as adding noise to downsampled HiC data, ensembling downsampled inputs, and retraining with real-world low-quality input datasets. We find that training with a real-world low-quality dataset substantially improves the performance of the deep learning model (an improvement of 84% in the similarity score).

[1] Synonymous with improving HiC quality, which subsequently increases the HiC resolution

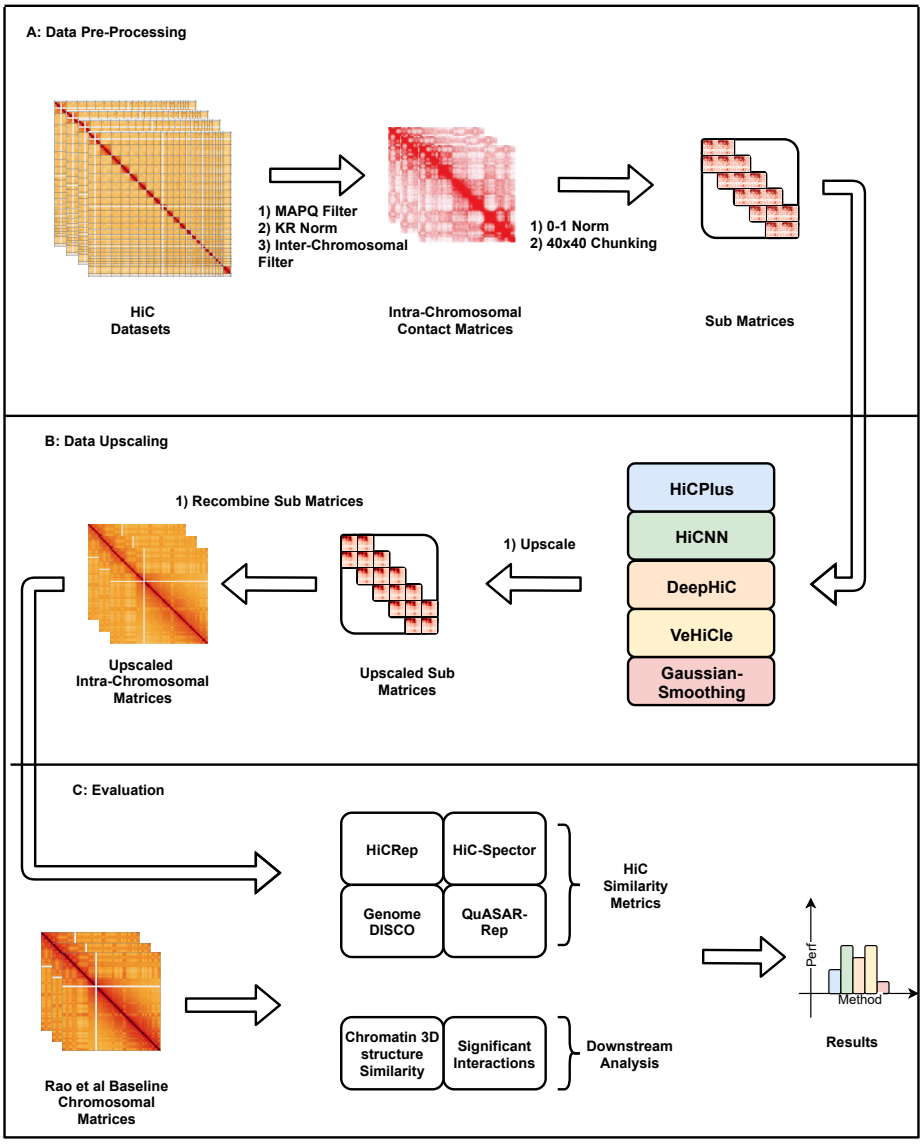

Figure 1: **Overview of our benchmarking study: A** Data pre-processing pipeline - We filtered the HiC matrices with the same MAPQ value ($\geq$30) and normalized them with the same KR normalization algorithm to ensure a fair comparison. We removed inter-chromosomal contacts because of their extremely sparse nature and performed a 0-1 normalization on intra-chromosomal matrices to reduce the impact of extreme values. Finally, we cropped appropriately sized sub-matrices to ensure that input is in the correct format for each upscaling algorithm. **B** We upscaled the sub-matrices using a wide variety of deep learning-based upscaling models and then recombined them to form upscaled intra-chromosomal HiC matrices. **C** We used multiple HiC-based similarity metrics for evaluation of the upscaling methods as well as downstream analyses to provide a comprehensive set of results that we can use to analyze the performance of each upscaling model.

Additionally, we perform two downstream analyses - (1) recreating the 3D chromatin structure from upscaled HiC matrices and (2) calling significant interactions in the upscaled HiC matrices. We again observe that the models trained with real-world low-quality datasets can recover more biological information than those trained with downsampled datasets. However, our significant interaction analysis suggests that the deep-learning methods fail to generalize to the inputs with high sparsity, even with retraining. This result suggests scope for further improvements in training strategies, model architecture, and evaluation protocols to ensure the deep learning models' applicability to real-world HiC upscaling tasks. Nevertheless, our results and analyses indicate that deep learning-based models trained on real-world HiC datasets are better suited for the HiC upscaling task

than the current models trained on downsampled HiC maps. Lastly, our study highlights that even when trained with a representative dataset, the existing methods fail to generalize to the entire spectrum of sparsities. This observation calls for a need for rigorous evaluation strategies, including evaluation with datasets from different experiments, cell lines, and varying sparsities to ensure that the performance translates to real-world scenarios.

2 Methods

2.1 Data pre-processing

We collected the HiC datasets for our study from multiple experimental sources. Following the existing deep-learning studies [3, 4, 9, 10, 14], we used HiC matrices for the GM12878, IMR90, and K562 cell lines from Rao *et al.* [5] as our primary High Read Count (HRC) datasets. We define HRC datasets as HiC experiments that are high-quality and formulated as high-resolution matrices. We selected six additional low-quality, low-resolution, or Low Read Count (LRC) HiC experiments to test the performance of deep learning models on the real-world HiC upscaling task. Four of these HiC experiments are for GM12878 cell line (two from ENCODE experiment ID ENCSR382RFU, one file from ENCODE experiment ID ENCSR968KAY and another from GEO accession code GSM1551582), one for K562 cell line (GEO accession code GSM1551622), one for IMR90 cell line (GEO accession code GSM1551606). Note that the GEO accession datasets are experimental replicates of the original Rao *et al.* dataset and represent a subset of the HRC HiC matrix. Table 1 summarizes the attributes of each dataset. We chose the bin size of 10kb resolution, and MAPQ value of $>=30$ with KR normalization [7] to balance the HiC matrices and filtered out reads with low statistical confidence to remove spurious reads and experimental artifacts. We further filtered out inter-chromosomal, and X chromosome reads as done in previous works. Therefore, we have twenty-two files containing intra-chromosomal contact values in a dense matrix format after pre-processing each dataset.

We performed additional pre-processing specifically for the existing deep-learning-based models. Specifically, these models need the input matrices to be cropped and be of a particular size. Moreover, they perform better if the input, in addition to KR normalization, is also normalized to have values between 0 and 1 ([4]). Therefore, we pre-processed all the Hi-C matrices, chromosome-wise, by clamping all the values between 0 and the 99.95^{th} percentile value of their read counts. Then we divided all the read count values by 99.95^{th} percentile values to standardize the matrices. Next, we cropped these matrices into sub-matrices that deep-learning-based models can finally use to upscale. Like previous works, we obtained these sub-matrices from the 200 bin range around the diagonal of the HiC matrix, which contains most of the biological features of the HiC matrices [14]. We used the standard dataset splits used by the existing works. The test set contains chromosomes 19-22, the validation set contains chromosomes 9-12, and the remaining chromosomes are in the training set. We exclusively trained (or retrained) all models with GM12878 cell line datasets and prediction on K562 and IMR90 represents the cross-celltype predictions.

2.2 Deep learning-based HiC upscaling models

We ran several state-of-the-art deep learning-based HiC upscaling methods to provide a comprehensive set of performance results. We use the PyTorch implementations of these methods provided by the papers to ensure a fair comparison of the off-the-shelf version of each technique. Our selected deep-learning models can be divided into two broad categories, those that employed adversarial loss to optimize the weights and those that did not. HiCPlus [14] used a three-layer convolutional network to optimize with MSE loss. HiCNN [10] extended the HiCPlus to a 54 layer architecture that had residual network (ResNet) layers [2] which resulted in a stable model with faster training. HiCGAN [9], DeepHiC [4] and VeHiCLe [3] employed Generative Adversarial Networks

Dataset	Absolute Read Counts	Relative Read Counts	Source
GM12878-HRC-1	1,844,107,778	1	GEO (GSE63525)
GM12878-LRC-1	42,453,795	1/44	ENCODE (ENCSR382RFU)
GM12878-LRC-2	37,079,587	1/50	ENCODE (ENCSR382RFU)
GM12878-LRC-3	70,138,184	1/26	ENCODE (ENCSR968KAY)
GM12878-LRC-4	18,696,952	1/99	GEO (GSM1551582)
IMR90-HRC-1	735,043,093	1	GEO (GSE63525)
IMR90-LRC-1	75,193,876	1/10	GEO (GSM1551606)
K562-HRC-1	641,402,880	1	GEO (GSE63525)
K562-LRC-1	44,882,605	1/14	GEO (GSM1551622)

Table 1: **Summary of the datasets and their sources.** All experiments used Mbol enzyme and filtered fragments to be in the size range of 300-500 using SPRI beads.

(GANs). GANs jointly optimizes two models, a generator that produces inputs and a discriminator that tells fake inputs from real ones, to learn to create increasingly more realistic outputs. HiCGAN used the MSE loss and the discriminator loss to optimize the weights. DeepHiC made extensions over HiCGAN to introduce perceptual loss and total variation loss to force the models to generate outputs with sharper and realistic features. Lastly, VeHiCle made two further modifications to the DeepHiC. It replaced the perceptual loss with a domain-specific HiC loss by using an unsupervised model trained to generate HiC inputs. 2) It added an insulation loss that forces the model to learn the underlying biological structure (specifically topologically associated domains) to generate biologically informative HiC matrices.

For our evaluations, we retrained HiCPlus and HiCNN because both predicted raw contact counts rather than a contact map with values between [0, 1], which made these models' outputs incomparable and also possibly at a disadvantage against the other models [4]. Moreover, DeepHiC used model ensembling (a model for 1/16, 1/25, 1/50, and 1/100 downsampling ratio inputs) to get performance across the entire range of HiC input sparsities. To make the comparison fair, we trained an additional three versions of HiCNN and HiCPlus suited to upscale inputs with 1/25, 1/50, and 1/100 to ensure a fair comparison at inputs from multiple downsampling ratios. Note that we exclude HiCGAN [9], a Generative Adversarial Network (GAN) model, from our comparisons as we were unable to run the model on our selected datasets. For DeepHiC and VeHiCle, we used the provided weights. For as a baseline upscaling algorithm we added Gaussian smoothing with kernel size of $n=17$ and 2D Gaussian distribution with $\sigma_x=\sigma_y=7$. Lastly, we did not perform hyper-parameter tuning for model retraining to compare the available versions of these methods. For additional information on these methods, please refer to Supplementary Section S1.

2.3 Evaluation metrics

Recent work has shown that correlation metrics (such as Pearsons and Spearmans) fail to assign an accurate reproducibility score for HiC experiments due to their inability to account for the underlying data distributions (e.g., distance effect in HiC matrices) [13]. Most current HiC upscaling studies have used correlation metrics such as Pearsons, Spearman's, and Structural similarity index metric (SSIM) to evaluate their performance. Due to the limitations of the correlation-based metrics, we conduct our analysis primarily based on HiC similarity metrics. To evaluate the state-of-the-art deep learning models, we use all four HiC specific similarity metrics (GenomeDISCO, HiCRep, HiC-Spector, and QuASAR-Rep) designed to calculate the HiC contact map reproducibility scores. Each metric compares the "ground truth" original high-resolution HiC map and the upscaled HiC map generated from the deep learning methods to provide a similarity score. To conserve

space and keep the evaluation concise, we only show results on SSIM and GenomeDISCO in the main text, and results for the rest of the metrics are in the supplementary section. SSIM uses two images' luminance, contrast, and structure to calculate the similarity. Whereas, GenomeDISCO uses Graph random walks to smooth out the contact matrices at varying scales to compute concordance scores between the two input maps. For detailed information on these metrics (and other metrics), refer to Supplementary Section S2.

2.4 Downstream analysis strategies

We perform two downstream analyses to assess the quality of biological information that we can recover from the upscaled HiC matrices generated by the deep learning methods from LRC matrices. Although similarity scores produced by QuASAR-Rep, GenomeDISCO, HiC-Rep, and HiC-Spector, are at their core, driven by underlying biology, they never directly compare the biological content, which corresponds to the underlying structure which read counts approximate in the HiC matrices. Therefore, we set up two downstream evaluation techniques to ensure the comprehensiveness evaluation. First, we use 3DMax [11] to generate a 3D model of chromatin from the upscaled matrices. We use out-of-the-box parameters to construct the 3D models of the chromatin and compare them using the Template modeling score (TM-Score) to estimate the reconstruction accuracy. A higher accuracy indicates better biological information contained in the upscaled data. Secondly, we use FitHic [6] to obtain all the significant interactions from the upscaled matrices. We then use the Jaccard index to find the similarity between significant contacts in the original HRC matrix and the upscaled matrix. A higher Jaccard index at low p-value cutoffs implies high biological information in the recovered matrices.

3 Results

In this section, we have performed a comprehensive set of experiments to establish the applicability of existing methods in real-world scenarios and the validity of the current evaluation protocols. We have divided our results into three parts. 1) We replicate the evaluation setup of existing methods [4, 9, 10, 14] and compare the performance of correlation-based metrics against that of HiC similarity metrics. 2) We report the performance of the existing methods on real-world Low Read Count (LRC) datasets. 3) We conduct 3D chromatin reconstruction and Significant interaction Recall analysis to assess the quality of biological information retrievable from upscaled matrices to establish their relevance for downstream applications.

3.1 DeepHiC and HiCNN give best performance for downsampled input HiC datasets across different downsampling fractions

We have set up a pipeline similar to DeepHiC, which dictates the downsampling ratios ($\frac{1}{16}$ to $\frac{1}{100}$), evaluation metrics, and cell lines (GM12878, IMR90 and K562) we have used to evaluate the performance of HiC upscaling models. In this experiment we explore our hypothesis that correlation based metrics (such as SSIM) can potentially lead to false conclusions. We condense our results in the Figure 2 where we show measurements on GenomeDISCO and SSIM. Results for the remaining metrics are in the Supplementary Tables S1, S2, S3, S4.

In the Figure 2 we show the upscaling performance (y-axis either SSIM or GenomeDISCO) at different sparsity ratios (x-axis) for the downsampled datasets. Based on the results, while all of these methods provide good HiC upscaling performance on lower downsampling ratios, only HiCNN and DeepHiC show similar or comparable performance on higher downsampling ratios which provides empirical proof that they generalize better on the downsampled datasets. Moreover, we observe that VeHiCLe shows consistent performance but worse than both HiCNN and DeepHiC, which is expected given VeHiCle was trained to upscale real-world LRC HiC matrices. While SSIM suggests that both VeHiCle and HiCPlus perform better than Gaussian smoothing,

GenomeDISCO implies an opposite conclusion. Thus confirming our hypothesis that SSIM (a correlation based metric) can lead to false conclusions. Considering these insights, we make two key choices for the rest of our evaluations. 1) We only consider HiCNN and DeepHiC as our prime candidates for retraining since they provide the best generalizability. 2) We only consider the results from the HiC specific similarity metrics when comparing the performance of HiC upscaling methods.

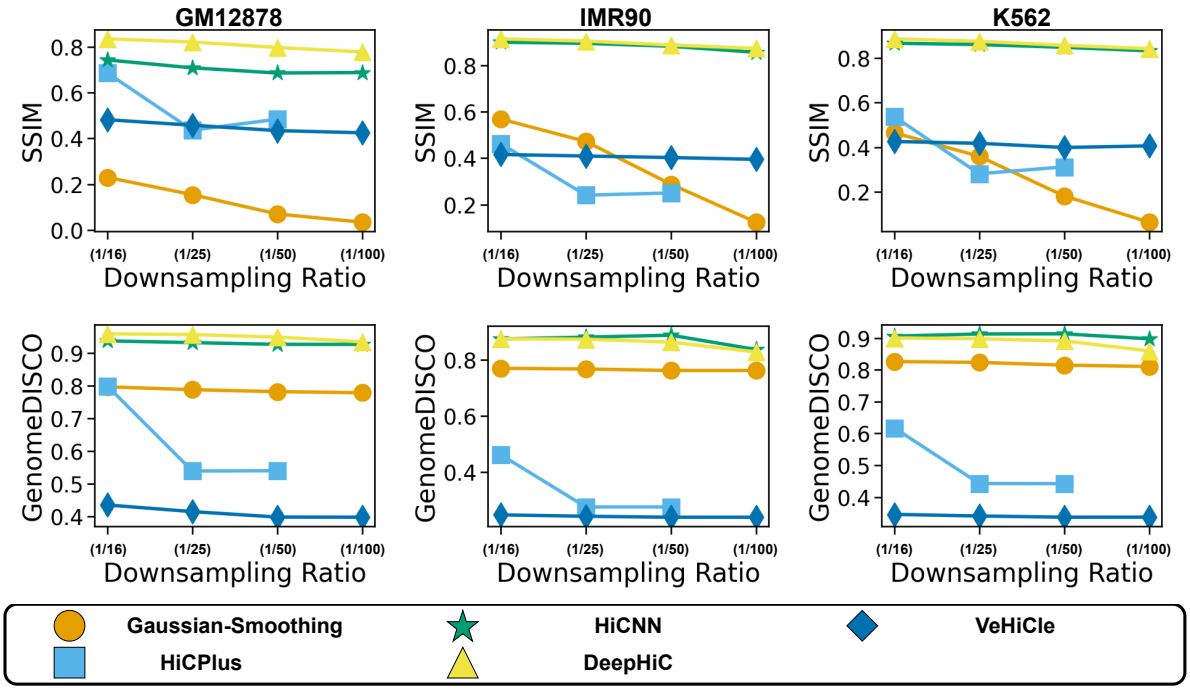

Figure 2: **Performance evaluation of deep learning-based methods for upscaling downsampled input HiC datasets**. On the x-axis, we have the downsampling ratio, and on the y-axis, we have either SSIM or GenomeDISCO. We observe as the downsampling ratio increases, only DeepHiC and HiCNN show similar or comparable performance. The figure also presents a discrepancy between the trends observed in SSIM and GenomeDISCO. SSIM suggests that both HiCPlus and VeHiCle show better performance than Gaussian Distribution, while GenomeDISCO suggests otherwise. This discrepancy highlights the limitations of the correlation/image-based metrics for evaluating HiC sample similarity and their potential to lead to false conclusions.

3.2 Investigating the performance of deep-learning methods on real-world LRC datasets

We conduct a series of tests to establish whether the existing HiC upscaling methods can achieve similar or comparable performance on real-world HiC matrices. We hypothesize that downsampled HiC matrices do not have the same underlying distribution as the real-world LRC matrices because the uniform downsampling of the read counts fails to incorporate the impact of experimental noise and consequently generates unrealistic HiC matrices. Thus we expect that models trained with downsampled datasets will not perform well on real-world LRC matrices.

In this evaluation phase, we investigate the following. (1) Is there a difference in the underlying distribution of real-world LRC datasets and the downsampled HRC datasets? (2) What is the impact of these differences on the performance of the models trained with downsampled datasets? (3) What retraining strategies can potentially improve the generalizability of the existing methods to real-world LRC datasets?

3.2.1 Distribution of downsampled HRC HiC input is different from the real-world LRC datasets

As a preliminary analysis of the data, we compare the distributions of both downsampled and real-world LRC HiC matrices. To do that, we calculate the similarity of the uniformly downsampled and the real-world LRC HiC matrices with the original HRC matrices from Rao et al. [5]. We visualize our results in a scatter plot in Figure 3 for all three cell lines. We plot the relative read count (analogous to sparsity ratio) on the x-axis and the GenomeDISCO score on the y-axis. We plot a line of best fit between the data points corresponding to the downsampled datasets (red circles) to establish a point of comparison for the real-world LRC datasets (green squares). The higher value corresponds to a higher correlation with the HRC dataset. If the distribution of the real-world LRC Hi-C matrices were similar to the downsampled datasets, we would expect their similarity scores to be close to the ones for the downsampled matrices. Therefore, the green points representing the real-world LRC datasets should either be on the line or be within a short distance to the best fit line to have the same distribution as the downsampled datasets.

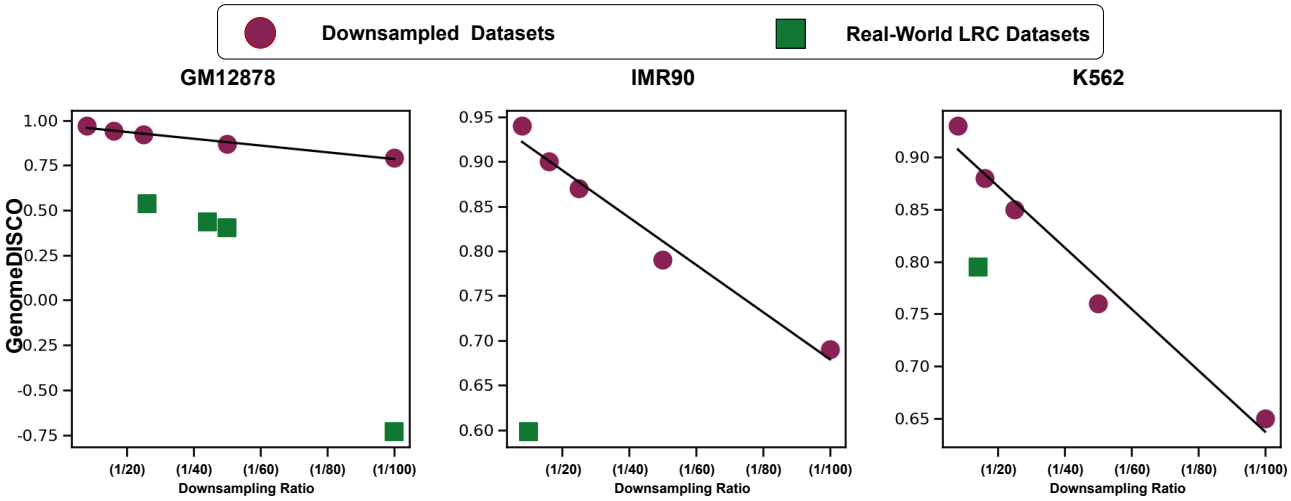

Figure 3: **Comparing the distributions of downsampled HiC datasets with the real-world LRC datasets.** The figure shows a relative read count ratio (or sparsity) on the x-axis and a GenomeDISCO score on the y-axis. We plot a line of best fit across the points of the downsampled datasets to establish a baseline of how the score should decrease with an increase in relative read count ratio. Our graphs show that real-world LRC points are always below that best-fit line, implying a higher score degradation for the same relative read count ratio. This observation suggests that the real-world LRC datasets follow a different distribution from the downsampled datasets.

We observe a large difference between the GenomeDISCO score of downsampled datasets and the real-world LRC datasets. Real-world LRC datasets yield lower GenomeDISCO scores, suggesting they are less similar to the HRC HiC datasets. For example, for $\frac{1}{100}$ relative read count, the downsampled dataset has a GenomeDISCO score of 0.8214 compared to -0.7317 for the LRC dataset. Therefore, the downsampled datasets do not sufficiently represent the underlying data distribution in LRC datasets. Moreover, as the read count ratio decreases, the distance between the line of best fit for downsampled datasets and the LRC dataset increases; this observation suggests that experimental noise is amplified non-linearly with respect to the read count ratio.

This analysis confirms our initial hypothesis that uniformly downsampling read counts fail to produce HiC matrices with identical information content as the real-world LRC matrices. Consequently, we anticipate that the models trained with these downsampled inputs will perform poorly on real-world LRC datasets. Lastly, we expect the performance difference to be more considerable on matrices with higher sparsity because they have higher experimental noise content.

3.2.2 Deep learning models trained on downsampled HiC input perform worse on real-world LRC datasets

We showed a steep difference in the distribution of the real-world LRC and the downsampled datasets. To investigate the impact of that difference, we evaluate the performance of the models trained with the downsampled HiC matrices (Baseline version) on real-world LRC matrices. We run these models on six real-world datasets categorized based on their relative read count ratios. Figure 4 presents the performance results for the GenomeDISCO metric.

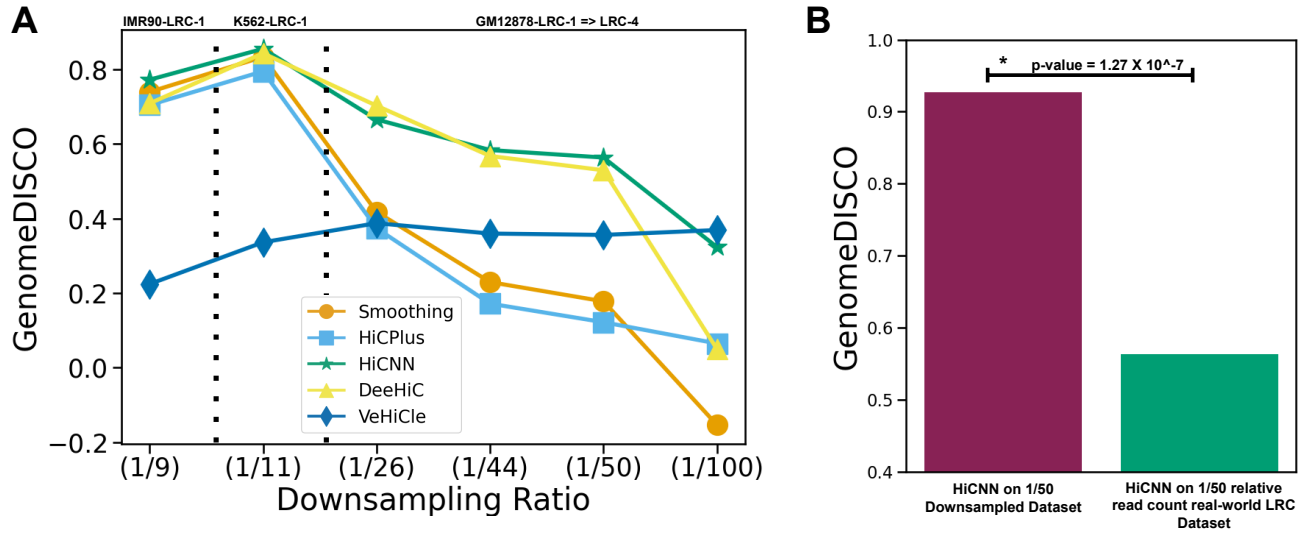

Figure 4: **Performance comparison for upscaling real-world LRC HiC datasets.** **(A)**: We see that the performance of the deep learning models degrades as the read counts of the input dataset decrease. **(B)**: The performance of HiCNN (best performing model on average) shows a substantial decrease in performance on LRC datasets compared to the downsampled datasets of the same ($\frac{1}{50}$) relative read count ratio

We observe in Figure 4(A) that HiCNN offers the best performance in four out of six datasets. This result suggests that HiCNN is an attractive choice for an out-of-the-box deep learning model to upscale real-world LRC datasets. However, the performance of most of these methods degrades substantially as the sparsity of the input dataset increases suggesting poor generalizability on the inputs with low read counts. The only exception is VeHiCle, which shows consistent performance across the downsampling ratios. This observation provides additional evidence supporting our hypothesis that deep-learning-based methods trained with downsampled inputs are sub-optimal for upscaling real-world LRC matrices. We additionally compare the performance of HiCNN on downsampled datasets against its performance on real-world LRC datasets in Figure 2(B) with similar relative read counts. On a real-world LRC dataset with the same relative read counts, we see a substantial drop in performance ($p\text{-}value = 1.27 \times 10^{-7}$). These observations are similar across all HiC specific evaluation metrics (see Supplementary Table S5). We see that HiCNN is the best performing model for 5/6 datasets for HiCRep, 5/6 for HiC-Spector, and 4/6 for QuASAR-Rep. Therefore, in the subsequent section, we explore potential retraining strategies for HiCNN to create robust models that are more efficient at upscaling real-world LRC datasets and possibly achieve performance similar to those reported on downsampled datasets by the existing studies.

3.2.3 Retraining deep learning models with real-world LRC datasets improves performance

To determine the effects of different retraining strategies on the generalizability of the HiCNN model on real-world datasets, we retrain the HiCNN with the following inputs: (1) Downsampled data which is augmented

with three different types of noise (Gaussian, Random, and Uniform); (2) a real-world LRC dataset; (3) ensemble of downsampled datasets with different downsampling ratios; (4) ensemble of real-world LRC datasets with different read counts. The critical insight behind all of these retraining strategies is to expose the model to a more realistic distribution in the training phase to improve the generalizability and performance in the evaluation phase. Most of the retraining strategies provided only modest improvements (ranging from an average decrease of 24 % for Gaussian-Noise in GenomeDISCO score to a 3 % improvement with the ensemble of downsampled inputs) over the baseline (Supplementary Table S6).. In Figure 5 **A** we observe that HiCNN-LRC (HiCNN retrained with a GM12878-LRC-1 dataset) outperforms the baseline HiCNN trained with downsampled datasets for 5/6 input LRC datasets. This result provides clear evidence that retraining with the LRC dataset improves performance and generalizability on real-world LRC datasets.

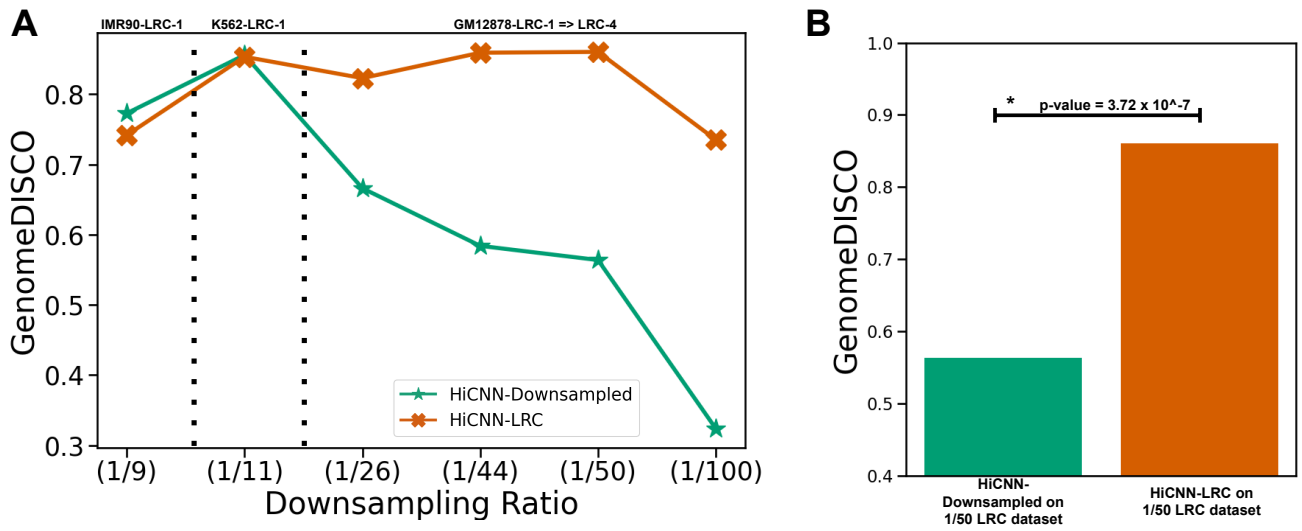

Figure 5: **Retraining HICNN on real-world LRC datasets. (A)**: We compare the performance of the baseline HiCNN (HiCNN-Downsampled) model with that of the HiCNN model retrained with real-world LRC datasets (HiCNN-LRC). The results show that HiCNN-LRC performs better and generalizes better to inputs with low relative read count ratios. **B**: To elaborate on the performance improvement, we show that HiCNN-LRC shows substantial improvement (with *p-value* $=1.02\times10^{-6}$ in performance over HiCNN-Downsampled on 1/50 relative read count real-world dataset.

To further emphasize the performance improvement, we compare the performance of the original HiCNN model against the performance of the retrained HiCNN model in Figure 5 (B). The bar graph shows a significant difference in GenomeDISCO scores of both of these models (*p-value* $=3.27\times10^{-7}$). We observe similar trends and observations on the other HiC specific similarity metrics, although the trend is not as clear for HiC-Spector and HiCRep. Regardless, the data suggests that the retrained models show performance improvement.

Zooming in on the results for GenomeDISCO, the dataset with very high relative read counts (IMR90-LRC-1) is the only case where HiCNN retrained on the real-world LRC dataset does not perform well. This observation ties us back to our previous hypothesis that baseline methods are equipped better to upscale inputs with small or no experimental noise. Consequently, this result implies space for further improvement in the training strategy (or the model architecture) to increase the robustness of the deep learning models against HiC inputs with varying sparsities. Summing it all up, we find that retraining deep-learning methods with real-world datasets improves their generalizability and performance in real-world scenarios.

3.3 Downstream Analysis

To determine whether the upscaled Hi-C matrices retain important biological signals, we analyze the 3D structure of chromatin and significant interactions reconstructed from the upscaled matrices. Unlike previous works [4, 9, 10, 14], we perform downstream analysis on upscaled real-world LRC datasets to provide insights into the applicability of these methods in real-world scenarios.

3.3.1 Retrained Models generate highly similar 3D structure of chromatin

We use the upscaled HiC maps produced for real-world LRC input matrices for all the cell lines and use the 3DMax tool [11] to construct the 3D structure for the 200×200 HiC sub-matrices for all of the sub-matrices in the test chromosomes. We use TM-Score to compute similarity between the 3D structures. The authors of VeHiCle perform a similar analysis [3], where they compare the recovered 3D structure of the chromatin from the upscaled HiC matrices to establish that their upscaled matrices are biologically informative. We summarize the analysis results in Table 2. We observe a similar trend to our previous evaluations on the HiC similarity metrics. HiCNN-LRC (HiCNN retrained with LRC inputs) performs better on the datasets with high sparsities, while the model trained with downsampled inputs performs better on low sparsity inputs. HiCNN-LRC performs the best in four out of the six cell lines. This observation implies that retraining HiCNN with LRC matrices improves the recovery of biological content in lower read count matrices. Hence further cementing our claim and suggestion for retraining upcoming (or existing) HiC upscaling methods with the real-world dataset.

	Baseline	Smoothing	HiCPlus	HiCNN	DeepHiC	VeHiCle	HiCNN-LRC
GM12878-LRC-1	0.3087	0.4593	0.4433	0.6126	0.6018	0.4573	**0.7748**
GM12878-LRC-2	0.2997	0.4292	0.4018	0.6111	0.5414	0.4353	**0.7541**
GM12878-LRC-3	0.3458	0.5048	0.4835	0.6481	0.7566	0.4846	**0.8186**
GM12878-LRC-4	0.3148	0.2854	0.3356	0.4913	0.3116	0.4136	**0.5652**
IMR90-LRC-1	0.2887	0.5807	0.5633	**0.6583**	0.5723	0.4142	0.6076
K562-LRC-1	0.3345	0.6523	0.6381	**0.7133**	0.6668	0.4823	0.6858

Table 2: **TM scores of HRC Chromatin structures vs upscaled and LRC Chromatin structures**. The best score for each dataset is bolded. HiCNN-LRC shows the best performance in four out of six datasets and has comparable performance in the two lower sparsity datasets.

3.3.2 Retrained model tends to recover a higher number of significant interactions

In the last experiment, we compare the number of significant interactions recovered using FitHiC [6] at varying *p-value* cutoffs to estimate useful biological information. For the interaction recovery curves, the higher the Jaccard index at lower *p-values* the better the information recovery. In Figure 6 we show that HiCNN-LRC (retrained with real-world LRC datasets) has the best curve in three (GM12878-LRC-1, GM12878-LRC-2, and IMR90-LRC-1) out of six datasets. Moreover, HiCNN-LRC shows a better Significant interaction recall curve than baseline HiCNN in 5 out of 6 datasets, supporting our suggestion to retrain the models with real-world LRC

datasets. However, none of the deep-learning-based models had a Jaccard Index greater than 10^{-2} for p-values greater than 0.0001 for the GM12878-LRC-4 dataset, implying that these models failed to learn the underlying biological structure for highly sparse input and ended up making spurious contact predictions. In conclusion, in conjunction with our previous evaluations, this result provides critical insight into these deep-learning models that they fail to generalize to the entire spectrum of inputs regardless of the training modality. This finding suggests a need for better training methods and alternative model architectures.

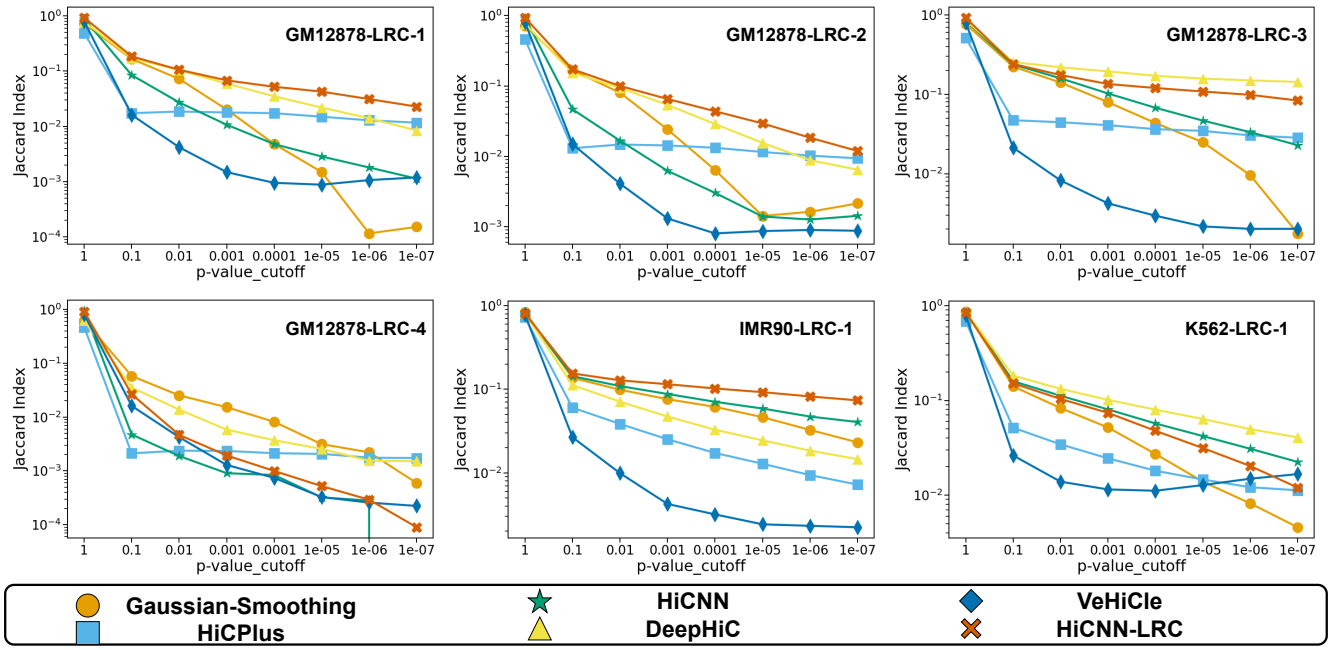

Figure 6: **Shows significant interaction profile at multiple p-value cutoffs.** On a high level, these graphs show that HiCNN-LRC yields better biological information recovery than other models in three out of the six datasets. However, on GM12878-LRC-4, none of the methods shows good information implying the need for methods that generalize to entire spectrum of input sparsity.

4 Discussion

This work has conducted a series of experiments to explore how the existing HiC upscaling methods perform in their intended real-world scenarios. Our results have strongly suggested that none of the proposed methods achieved similar or comparable performance. Regardless, from the existing techniques, HiCNN, a non-adversarial model, showed the best generalizability out of the box, hinting at potential overfitting in the nuanced methods towards HiC matrices of a particular modality. We suggest that the upcoming methods employ stricter evaluation protocols to ensure that the advertised performance would be comparable in real-world scenarios. We would still want to highlight that we neither evaluated these methods on single-cell datasets or HiC experiments treated with different experimental protocols. We hypothesize that the performance of these methods on those datasets would be worse.

Furthermore, in this study, we explored possible methods to improve generalizability through retraining with different modalities of input augmentation. Our analysis found that retraining HiCNN with the real-world low-read count HiC matrices improved the performance over the baseline HiCNN and its generalizability to inputs of varying levels of sparsities. However, we found that even with retraining, HiCNN failed to generalize to the entire spectrum of sparsity-levels. Although in this work, we do not retrain any of the more nuanced methods

such as DeepHiC or VeHiCle, we still believe that the existing methods could benefit from our results and further improve their setup to learn correct and more meaningful latent representations of the real-world data.

References

[1] J. R. Dixon, S. Selvaraj, F. Yue, A. Kim, Y. Li, Y. Shen, M. Hu, J. S. Liu, and B. Ren. Topological domains in mammalian genomes identified by analysis of chromatin interactions. *Nature*, 485(7398):376–380, 2012.

[2] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition, 2015.

[3] M. Highsmith and J. Cheng. Vehicle: A variationally encoded hic loss enhancement algorithm for improving and generating hi-c data. *Scientific Reports*, 11(1), 2021.

[4] H. Hong, S. Jiang, H. Li, G. Du, Y. Sun, H. Tao, C. Quan, C. Zhao, R. Li, W. Li, and et al. Deephic: A generative adversarial network for enhancing hic data resolution. *PLOS Computational Biology*, 16(2), 2020.

[5] F. Jin, Y. Li, J. R. Dixon, S. Selvaraj, Z. Ye, A. Y. Lee, C.-A. Yen, A. D. Schmitt, C. A. Espinoza, B. Ren, and et al. A high-resolution map of the three-dimensional chromatin interactome in human cells. *Nature*, 503(7475):290–294, 2013.

[6] A. Kaul, S. Bhattacharyya, and F. Ay. Identifying statistically significant chromatin contacts from hic data with fithic2. *Nature Protocols*, 15(3):991–1012, 2020.

[7] P. A. Knight and D. Ruiz. A fast algorithm for matrix balancing. *IMA Journal of Numerical Analysis*, 33(3):1029–1047, 2012.

[8] E. Lieberman-Aiden, N. L. van Berkum, L. Williams, M. Imakaev, T. Ragoczy, A. Telling, I. Amit, B. R. Lajoie, P. J. Sabo, M. O. Dorschner, R. Sandstrom, B. Bernstein, M. A. Bender, M. Groudine, A. Gnirke, J. Stamatoyannopoulos, L. A. Mirny, E. S. Lander, and J. Dekker. Comprehensive mapping of long-range interactions reveals folding principles of the human genome. *Science*, 326(5950):289–293, 2009.

[9] Q. Liu, H. Lv, and R. Jiang. hicGAN infers super resolution Hi-C data with generative adversarial networks. *Bioinformatics*, 35(14):i99–i107, 07 2019.

[10] T. Liu and Z. Wang. HiCNN: a very deep convolutional neural network to better enhance the resolution of Hi-C data. *Bioinformatics*, 35(21):4222–4228, 04 2019.

[11] O. Oluwadare, Y. Zhang, and J. Cheng. A maximum likelihood algorithm for reconstructing 3d structures of human chromosomes from chromosomal contact data. *BMC Genomics*, 19(1), 2018.

[12] Rao and et al. A 3d map of the human genome at kilobase resolution reveals principles of chromatin looping. *Cell*, 159(7):1665–1680, 2014.

[13] G. G. Yardımcı, H. Ozadam, M. E. Sauria, O. Ursu, K.-K. Yan, T. Yang, A. Chakraborty, A. Kaul, B. R. Lajoie, F. Song, and et al. Measuring the reproducibility and quality of hi-c data. *Genome Biology*, 20(1), 2019.

[14] Y. Zhang, L. An, J. Xu, B. Zhang, W. J. Zheng, M. Hu, J. Tang, and F. Yue. Enhancing hi-c data resolution with deep convolutional neural network hicplus. *Nature Communications*, 9(1), 2018.

Investigating the performance of deep learning methods for Hi-C resolution improvement

Supplementary Material

S1 Details of the evaluated methods

- **Gaussian Smoothing** We applied the Gaussian Smoothing filter, commonly used as a baseline [14], to establish emphasize the performance benefits of the deep-learning-based methods. This method uses a 2D kernel of shape $n \times n$ where n is a hyper-parameter. Each kernel value follows a 2D Gaussian distribution with hyperparameters σ_x and σ_y that represent the relative importance of neighboring features in prediction. The smoothing operation convolves this kernel on each pixel of a 2D image, or in our case, a HiC matrix read count, to update its value. This updated value contains the average of the neighboring values weighted by 2D Gaussian distribution in the kernel. This smoothing operation removes noise in the input matrix and improves the peak signal to noise ratio (PSNR) at the cost of blurring the features. For our experiments, we performed a grid search and found the kernel size of $n = 17$ and $\sigma_x = \sigma_y = 7$ to give the best reproducibility score on the validation set of LRC HiC matrices.

- **HiCPlus [14]** HiCPlus is the first application of deep learning to improve HiC resolution. It utilizes a standard three-layer convolutional neural network (CNN) architecture to upscale a low-resolution HiC matrix by mapping it to the target high-resolution matrix. To optimize its parameters, HiCPlus uses a mean squared error (MSE) loss. HiCPlus inputs HiC matrices as sub-matrices of size 40×40 binned at 10Kbp resolution and taken from 2 Mbp distance from the diagonal. All the following deep learning-based models follow the same input formulation. For this study, we make certain modifications to the original code to make HiCPlus comparable to the more recent implementations. For example, the original HiCPlus was trained to generate raw read counts rather than normalized HiC matrices. Therefore, we retrain the model to work with normalized high-resolution and low-resolution pairs of HiC sub-matrices. Moreover, the original implementation was trained to upscale only the matrices that had been downsampled to $\frac{1}{16}$ of the read counts of a high-resolution HiC map. We retrain the HiCPlus with three additional input HiC datasets with $\frac{1}{25}$, $\frac{1}{50}$, and $\frac{1}{100}$ downsampling ratios to explore the performance on matrices beyond the original downsampled version.

- **HiCNN [10]** HiCNN model also uses a CNN architecture like HiCPlus, except that it consists of a much deeper 54-layer neural network with ResNet layers [2]. The choice of ResNet layers provides two key benefits - (1) it provides additional architectural complexity to learn relevant non-linear relationships between the inputs and the outputs, and (2) they train significantly faster than regular CNN layers saving substantial time during training. In addition, the skip connections in ResNet layers further avoid model overfitting on the data. HiCNN, similar to HiCPlus, uses an MSE loss to optimize its parameters and learn the mapping between the input low-resolution matrix and the high-resolution target matrix. It also produces raw read counts and is trained with downsampling ratios of $\frac{1}{8}$ and $\frac{1}{16}$ and $\frac{1}{25}$. Therefore, for consistency, we retrain the model with additional datasets with downsampling rations of $\frac{1}{50}$ and $\frac{1}{100}$ and standardized values.

- **HiCGAN [9]** HiCGAN paper argues that the Mean Squared Loss function used in both HiCNN and HiCPlus causes these models to generate over-smooth matrices. Therefore, it proposes using a Generative Adversarial Network (GAN) model for the HiC resolution improvement task. A GAN architecture consists of (1) a generator and (2) a discriminator. The generator's objective is to produce data that increasingly

resembles the original distribution, and the goal of the discriminator is to identify fake (generated) data from the original data. This coupled training causes both models to get iteratively better at their tasks. HiCGAN uses a specialized form of GAN, which is called the conditional GAN (cGAN). In cGANs the generator produces an output conditional on the provided input instead of a random noise input. To optimize the parameters in the model, HiCGAN uses the discriminator loss and the MSE loss to generate matrices that are highly similar to the target HiC matrices. The HiCGAN paper shows that this method produces better-quality HiC matrices with sharper and more prominent features than the previous method.

- **DeepHiC [4]** DeepHiC paper, like HiCGAN [9], argues that the Mean Squared Loss function used in both HiCNN and HiCPlus causes these models to generate over-smooth matrices. Therefore, it uses Generative Adversarial Network (GAN) model for the HiC resolution improvement task. A GAN architecture consists of (1) a generator and (2) a discriminator. The generator's objective is to produce data that increasingly resembles the original distribution, and the goal of the discriminator is to identify fake (generated) data from the original data. This coupled training causes both models to get iteratively better at their tasks. DeepHiC substantially revises the previously proposed loss functions to contain additional functions that include Total Variation Loss and Perceptual Loss. These loss function along side Mean Squared Error Loss and Discriminator Loss causes the model to generate matrices with sharper features that are more biologically informative [4]. The paper also shows that training the deep learning models on standardized HiC matrices improves their performance further. Moreover, the paper trains the DeepHiC model with a downsampling ratio of up to $\frac{1}{100}$ to have model weights available for even the sparsest real-world HiC matrices.
- **VeHiCLE [3]** VeHiCLE is another GAN-based model like HiCGAN and DeepHiC. However, it makes additions to both the model architecture and loss functions used while training. Apart from using a GAN architecture, it also trains a variational auto-encoder (VAE). The output obtained by passing the HiC matrices through the trained VAE is used in a loss function to train the GAN. This loss obtained from the VAE is called the variation loss. VeHiCLE also uses the adversarial and MSE losses seen in previous methods. However, it introduces yet another loss called insulation loss. The insulation loss is a biologically inspired loss that utilizes insulation scores used to identify TADs in a HiC contact matrix. VeHiCLE is trained on input and target matrices of sizes 269×269, unlike the previous methods that used 40×40 sub-matrices as inputs. The paper shows that this increased matrix size improves performance, thus hypothesizing that the 40×40 matrices are too small to adequately capture information about large-scale HiC features (such as TADs). We created new datasets for VeHiCle that had HiC sub-matrices of size 269x269 to ensure a fair comparison of VeHiCle with other deep learning based methods.

S2 Details of the evaluation metrics

- **Mean Squared Error (MSE)** Mean Squared Error or MSE is the squared differences of the errors between the predicted value and the true/actual value. Smaller Mean Squared error is representative of better reproduction of the signal and consequently better quality of the predicted signal.
- **Mean Average Error (MAE)** Similar to Mean Square Error, Mean Average Error is the average of the differences of errors (rather than squared) between the predicted value and the true/actual value. MAE value of zero corresponds to an excellent signal reproduction.
- **Peak Signal to Noise Ratio (PSNR)** Peak Signal to Noise Ratio or PSNR is the ratio of maximum possible value or power of the signal to the power of the noise in the signal. PSNR is generally calculated in Decibels, a logarithmic scale, to compensate for the wide range of values the ratio can take. The higher the ratio value in decibels, the better the quality of the signal with respect to the baseline signal.

- **Structural Similarity Index Measure (SSIM)** Structural Similarity Index Measure (SSIM) is a metric that measures the perceived perceptual quality of an image against an original undistorted and higher quality image. SSIM measures this perceptual quality by comparing the luminance, contrast, and structural properties in small local regions of the images. A weighted sum of these properties allows SSIM to assign a similarity score that closely mimics the way humans perceive differences in images. However, we postulate that the HiC contact maps should be compared based on their underlying biological properties instead of their visual similarities. Therefore, assigning a similarity score based on SSIM score may hold little biological relevance and might lead to misleading conclusions about the quality of the generated datasets.

- **Pearson Correlation Coefficient (PCC)** Pearsons Correlation Coefficient (PCC) is a linear measure of the correlation between two sets of data distribution. PCC is a ratio of covariance between two variables and its product with their standard distribution. This metric essentially measures covariance between the two datasets, normalized to have a value between -1 and 1. Here, -1 or 1 values imply highly negatively or positively correlated, respectively, and a 0 value implies no correlation between the two datasets.

- **Spearman's rank Correlation Coefficient (SCC)** Spearman's rank Correlation Coefficient (SCC) measures the statistical dependence between the rank of two variables. This measure essentially captures how well two variables can be described using a monotonic function. SCC between two variables is equal to the PCC of the rank of variables. Thus, SCC has a value of +1 or -1 when either of the variables is a perfect monotone of the other. It has a value of 0 when they do not correlate monotonically.

S3 Results

S3.1 Deep-learning methods perform well on downsampled input HiC datasets

	MSE	MAE	PSNR	SSIM	PCC	SCC	HiCRep	HiC-Spector	GenomeDISCO	QuASAR-Rep
GM12878										
Baseline	0.1101	0.2096	5.5260	0.0629	0.5449	0.7333	0.9500	0.4199	0.8882	0.5581
Smoothing	0.0532	0.1519	11.0475	0.2301	0.8895	0.9088	0.8173	0.3801	0.7969	0.8191
HiCPlus	0.0032	0.0346	24.2552	0.6858	0.9543	0.9191	0.9643	0.4908	0.7992	0.8808
HiCNN	0.0064	0.0403	20.8171	0.7424	0.9828	0.9191	0.9645	0.6433	0.9376	0.8884
DeepHiC	**0.0007**	**0.0162**	**31.8343**	**0.8346**	**0.9889**	**0.9230**	**0.9683**	**0.6927**	**0.9593**	**0.9049**
VeHiCle	0.0107	0.0542	19.6362	0.4817	0.8654	0.7721	0.7871	0.2143	0.4348	0.6376
IMR90										
Baseline	0.0374	0.0888	6.6400	0.2899	0.6867	0.6391	0.9100	0.3952	0.8707	0.4289
Smoothing	0.0120	0.0483	16.9545	0.5684	0.8745	0.8451	0.7759	0.3136	0.7708	0.7594
HiCPlus	0.0044	0.0436	22.7513	0.4622	0.9026	0.8459	0.9461	0.3139	0.4619	0.7681
HiCNN	0.0013	0.0129	27.9700	0.8985	0.9672	**0.8555**	0.9480	0.4895	0.8757	0.8174
DeepHiC	**0.0004**	**0.0101**	**33.6539**	**0.9131**	**0.9846**	0.8552	**0.9558**	**0.5327**	**0.8763**	**0.8222**
VeHiCle	0.0082	0.0506	17.8966	0.4155	0.7346	0.7077	0.7017	0.1763	0.2467	0.5635
K562										
Baseline	0.0415	0.1029	4.5044	0.1743	0.7238	0.6370	0.8877	0.4072	0.8869	0.3120
Smoothing	0.0149	0.0592	14.6678	0.4654	0.8807	0.8493	0.7960	0.3406	0.8265	0.7226
HiCPlus	0.0040	0.0408	22.5636	0.5373	0.9028	0.8555	0.9479	0.3517	0.6174	0.7589
HiCNN	0.0017	0.0148	25.5845	0.8659	0.9532	0.8593	0.9432	0.5031	**0.9064**	0.7818
DeepHiC	**0.0005**	**0.0119**	**33.1275**	**0.8857**	**0.9806**	**0.8625**	**0.9462**	**0.5300**	0.9007	**0.8013**
VeHiCle	0.0084	0.0496	16.1977	0.4262	0.7024	0.7215	0.7105	0.1874	0.3454	0.5278

Table S1: Results on 1/16 downsampled datasets. DeepHiC performs better than all the other methods across all three cell lines. Results from HiC Specific similarity metrics and Correlation metrics imply the same results

	MSE	MAE	PSNR	SSIM	PCC	SCC	HiCRep	HiC-Spector	GenomeDISCO	QuASAR-Rep
GM12878										
Baseline	0.1438	0.2501	1.4953	0.0399	0.4977	0.6692	0.9258	0.3849	0.8994	0.4404
Smoothing	0.0801	0.1940	8.7377	0.1547	0.8588	0.9068	0.8136	0.3787	0.7881	0.8144
HiCPlus	0.0076	0.0728	19.8004	0.4361	0.9396	0.8008	0.8882	0.3346	0.5389	0.7823
HiCNN	0.0083	0.0452	19.3186	0.7082	0.9817	0.9137	0.9596	0.6068	0.9324	0.8774
DeepHiC	**0.0008**	**0.0170**	**31.2796**	**0.8207**	**0.9872**	**0.9171**	**0.9636**	**0.6455**	**0.9571**	**0.8899**
VeHiCle	0.0141	0.0594	18.0991	0.4577	0.8350	0.7643	0.7747	0.2127	0.4145	0.6190
IMR90										
Baseline	0.0486	0.1067	5.4513	0.2427	0.6300	0.5697	0.8770	0.3576	0.8563	0.3360
Smoothing	0.0176	0.0641	15.1187	0.4715	0.8858	0.8414	0.7712	0.3093	0.7682	0.7552
HiCPlus	0.0119	0.1026	17.6353	0.2405	0.8890	0.6283	0.7814	0.2249	0.2750	0.6545
HiCNN	0.0012	0.0126	28.0559	0.8938	0.9648	**0.8465**	0.9396	0.4702	**0.8821**	**0.8035**
DeepHiC	**0.0005**	**0.0106**	**33.1092**	**0.9034**	**0.9814**	0.8453	**0.9465**	**0.4850**	0.8748	0.8012
VeHiCle	0.0092	0.0512	15.7957	0.4089	0.6935	0.6976	0.6846	0.1658	0.2416	0.5438
K562										
Baseline	0.0547	0.1244	2.7874	0.1311	0.6623	0.5685	0.8500	0.3773	0.8737	0.2300
Smoothing	0.0224	0.0796	12.7079	0.3598	0.8918	0.8460	0.7937	0.3347	0.8237	0.7157
HiCPlus	0.0112	0.0982	17.5974	0.2819	0.8881	0.6426	0.8264	0.2810	0.4425	0.6409
HiCNN	0.0016	0.0148	25.7195	0.8601	0.9519	0.8503	0.9343	0.4836	**0.9132**	0.7634
DeepHiC	**0.0006**	**0.0125**	**32.4625**	**0.8745**	**0.9767**	**0.8533**	**0.9354**	**0.4897**	0.8981	**0.7753**
VeHiCle	0.0092	0.0505	14.7026	0.4182	0.6679	0.7113	0.7022	0.1838	0.3403	0.5109

Table S2: Results on 1/25 downsampled datasets. DeepHiC performs better than all the other methods except in IMRR90 where HiCNN performs better on GenomeDISCO and QuASAR-Rep metric. On this downsampling the agreement between the correlation metrics and HiC specific similarity metrics seems to be diverging.

	MSE	MAE	PSNR	SSIM	PCC	SCC	HiCRep	HiC-Spector	GenomeDISCO	QuASAR-Rep
GM12878										
Baseline	0.2283	0.3444	-2.2773	0.0184	0.4056	0.5603	0.8727	0.3419	0.8764	0.2746
Smoothing	0.1551	0.2908	5.3602	0.0705	0.7640	0.8997	0.8070	0.3763	0.7822	0.8031
HiCPlus	0.0124	0.0656	15.9827	0.4839	0.8689	0.6305	0.8741	0.2717	0.5398	0.6828
HiCNN	0.0093	0.0476	18.5767	0.6855	0.9791	0.9049	0.9461	0.5508	0.9270	**0.8566**
DeepHiC	**0.0011**	**0.0189**	**30.2157**	**0.7970**	**0.9834**	**0.9061**	**0.9512**	**0.5960**	**0.9493**	0.8656
VeHiCle	0.0177	0.0643	16.5515	0.4342	0.8014	0.7536	0.7606	0.1989	0.3978	0.5937
IMR90										
Baseline	0.0817	0.1535	3.7817	0.1597	0.5160	0.4656	0.7992	0.3155	0.7823	0.2244
Smoothing	0.0408	0.1122	11.1264	0.2878	0.8798	0.8319	0.7643	0.3060	0.7628	0.7342
HiCPlus	0.0105	0.0842	16.8884	0.2500	0.7801	0.4823	0.7692	0.1895	0.2750	0.6084
HiCNN	0.0011	0.0129	28.7837	0.8812	0.9624	**0.8323**	0.9162	**0.4319**	**0.8891**	**0.7803**
DeepHiC	**0.0007**	**0.0119**	**31.8325**	**0.8865**	**0.9771**	0.8313	**0.9230**	0.4285	0.8643	0.7774
VeHiCle	0.0099	0.0519	14.6894	0.4023	0.6513	0.6852	0.6730	0.1616	0.2382	0.5254
K562										
Baseline	0.0940	0.1807	1.6881	0.0682	0.5355	0.4628	0.7652	0.3166	0.8123	0.1378
Smoothing	0.0518	0.1384	8.7968	0.1827	0.8817	0.8371	0.7844	0.3344	0.8151	0.6926
HiCPlus	0.0093	0.0756	16.7665	0.3126	0.7791	0.5028	0.8144	0.2266	0.4422	0.5711
HiCNN	0.0017	0.0154	25.6292	0.8477	0.9488	0.8375	**0.9110**	**0.4404**	**0.9137**	0.7362
DeepHiC	**0.0008**	**0.0138**	**31.3233**	**0.8565**	**0.9690**	**0.8395**	0.9059	0.4296	0.8914	**0.7420**
VeHiCle	0.0102	0.0547	14.3228	0.3994	0.6282	0.6981	0.6930	0.1627	0.3365	0.4838

Table S3: Results on 1/50 downsampled datasets. DeepHiC performs better on Correlation based metrics and GM12878 Cell line while HiCNN performs better on Biological metrics on IMR90 and K562 cell lines. In this evaluation table, there is a significant disagreement between the correlation metrics and HiC Specific similarity metrics

GM12878										
	MSE	**MAE**	**PSNR**	**SSIM**	**PCC**	**SCC**	**HiCRep**	**HiC-Spector**	**GenomeDISCO**	**QuASAR-Rep**
Baseline	0.3911	0.5050	-3.8888	0.0081	0.2922	0.4267	0.7963	0.2969	0.8214	0.1574
Smoothing	0.3108	0.4540	2.0528	0.0346	0.5932	0.8781	0.8024	0.3654	0.7787	0.7796
HiCPlus	-	-	-	-	-	-	-	-	-	-
HiCNN	0.0089	0.0465	18.5699	0.6874	0.9761	**0.8971**	0.9297	0.4995	0.9271	0.8326
DeepHiC	**0.0013**	**0.0206**	**29.2988**	**0.7773**	**0.9792**	0.8963	**0.9365**	**0.5194**	**0.9348**	**0.8352**
VeHiCle	0.0179	0.0646	16.9453	0.4245	0.7908	0.7469	0.7504	0.1848	0.3974	0.5818
IMR90										
Baseline	0.1603	0.2475	2.2718	0.0743	0.3732	0.3659	0.6939	0.2468	0.6285	0.1588
Smoothing	0.1070	0.2080	6.7167	0.1242	0.7950	0.8165	0.7562	0.3017	0.7629	0.7122
HiCPlus	-	-	-	-	-	-	-	-	-	-
HiCNN	0.0012	0.0140	28.6469	0.8550	0.9580	**0.8216**	**0.8862**	**0.3865**	**0.8378**	**0.7599**
DeepHiC	**0.0009**	**0.0133**	**30.6493**	**0.8718**	**0.9709**	0.8193	0.8859	0.3772	0.8293	0.7438
VeHiCle	0.0100	0.0528	16.1047	0.3946	0.6385	0.6770	0.6699	0.1529	0.2383	0.5151
K562										
Baseline	0.1922	0.2962	-0.5746	0.0215	0.3746	0.3591	0.6566	0.2494	0.6709	0.0889
Smoothing	0.1333	0.2525	5.2324	0.0653	0.7772	0.8197	0.7716	0.3193	0.8108	0.6616
HiCPlus	-	-	-	-	-	-	-	-	-	-
HiCNN	0.0016	0.0160	26.3740	0.8318	0.9463	0.8255	**0.8847**	0.3914	**0.8980**	**0.7108**
DeepHiC	**0.0009**	**0.0148**	**30.5163**	**0.8419**	**0.9635**	**0.8267**	0.8721	**0.3939**	0.8599	0.7032
VeHiCle	0.0100	0.0510	14.9243	0.4067	0.6149	0.6887	0.6898	0.1562	0.3367	0.4677

Table S4: Results on 1/100 downsampled datasets. DeepHiC performs better on Correlation based metrics and GM12878 Cell line while HiCNN performs better on Biological metrics on IMR90 and K562 cell lines. In this evaluation table, there is a significant disagreement between the correlation metrics and HiC Specific similarity metrics. Note: we have excluded the results of HiCPlus because it failed to produce meaningful upscaled matrices from 1/100 downsampled matrices

S3.2 Deep learning models trained on downsampled HiC input perform worse on real-world LRC datasets

ENCODE Datasets												
	HiCRep			**HiC-Spector**			**GenomeDISCO**			**QuASAR-Rep**		
	LRC-1	**LRC-2**	**LRC-3**	**LRC-1**	**LRC-2**	**LRC-3**	**LRC-1**	**LRC-2**	**LRC-3**	**LRC-1**	**LRC-2**	**LRC-3**
Baseline	0.7281	0.7081	0.8537	0.3019	0.2870	0.4066	0.4348	0.4043	0.5359	0.2026	0.1722	0.3287
Smoothing	0.7014	0.6779	0.7707	0.3270	0.3147	0.3574	0.2295	0.1778	0.4180	0.6994	0.6871	0.7352
HiCPlus	0.6503	0.6378	0.7610	0.3595	0.3558	0.3903	0.1717	0.1218	0.3740	0.5695	0.5339	0.6681
HiCNN	**0.9108**	**0.8973**	**0.9476**	**0.4517**	**0.4304**	0.5256	**0.5840**	**0.5638**	0.6661	**0.7708**	**0.7503**	0.8103
DeepHiC	0.9006	0.8828	0.9339	0.4253	0.3964	**0.5257**	0.5680	0.5297	**0.7030**	0.7542	0.7315	**0.8127**
VeHiCLe	0.6918	0.6794	0.7529	0.1785	0.1657	0.2075	0.3598	0.3561	0.3877	0.5229	0.5025	0.5934
Rao et. al Datasets												
	HiCRep			**HiC-Spector**			**GenomeDISCO**			**QuASAR-Rep**		
	GM12878	**IMR90**	**K562**	**GM12878**	**IMR90**	**K562**	**GM12878**	**IMR90**	**K562**	**GM12878**	**IMR90**	**K562**
Baseline	0.2883	0.5294	0.7674	0.0696	0.1449	0.3535	-0.7317	0.5982	0.7949	0.0732	0.1686	0.1896
Smoothing	0.4869	0.7003	0.8144	0.2412	0.2926	0.3454	-0.1539	0.7399	0.8341	**0.5498**	0.7041	0.6861
HiCPlus	0.5559	0.6076	0.7316	**0.3356**	0.2459	0.3607	0.0647	0.7049	0.7953	0.5213	0.5527	0.6034
HiCNN	0.6161	**0.7477**	**0.8984**	0.3260	**0.3196**	**0.4119**	0.3235	**0.7727**	**0.8558**	0.4953	**0.7384**	**0.7305**
DeepHiC	0.5396	0.6515	0.8829	0.2493	0.2724	0.3849	0.0496	0.7092	0.8437	0.4960	0.5031	0.7178
VeHiCLe	**0.6236**	0.5173	0.6914	0.1281	0.1255	0.1539	**0.3692**	0.2242	0.3364	0.4030	0.4803	0.4848

Table S5: Comparison of all the models trained on downsampled inputs on six real-world LRC datasets divided into two tables based on which HiC experiment they belong to. HiCNN outperforms all the other methods in 18 out of 24 combinations of datasets and HiC specific metrics. Results provide strong evidence that HiCNN generalizes better than all the other methods on the real-world datasets.

S3.3 Retraining deep learning models with real-world LRC datasets improves performance

ENCODE Datasets												
	HiCRep			HiC-Spector			GenomeDISCO			QuASAR-Rep		
	LRC-1	LRC-2	LRC-3	LRC-1	LRC-2	LRC-3	LRC-1	LRC-2	LRC-3	LRC-1	LRC-2	LRC-3
Downsampled -Best	0.9108	0.8973	**0.9476**	0.4517	0.4304	**0.5256**	0.5840	0.5638	0.6661	0.7708	0.7503	0.8103
Gaussian -Noise	0.8983	0.8832	0.9363	0.4132	0.3985	0.4953	0.3529	0.3051	0.5363	0.6519	0.6406	0.7120
Uniform -Noise	0.9062	0.8930	0.9342	0.4480	0.4252	0.5168	0.5022	0.4613	0.6475	0.7545	0.7394	0.8045
Random -Noise	0.9056	0.8926	0.9363	0.4517	0.4281	0.5161	0.4954	0.4475	0.6475	0.7534	0.7387	0.8050
Downsampled -Ensemble	0.8988	0.8915	0.9092	0.3839	0.3605	0.4669	0.5702	0.5402	0.6933	0.7268	0.6944	0.7932
Real-World -LRC	**0.9345**	**0.9316**	0.9345	**0.4717**	**0.4424**	0.4730	**0.8591**	**0.8603**	**0.8229**	**0.7823**	**0.7684**	**0.8192**
Real-World-LRC-Ensemble	0.9080	0.9233	0.8529	0.4400	0.4221	0.3637	0.7862	0.7804	0.8330	0.7427	0.7335	0.6900
Rao et. al Datasets												
	HiCRep			HiC-Spector			GenomeDISCO			QuASAR-Rep		
	GM12878	IMR90	K562	GM12878	IMR90	K562	GM12878	IMR90	K562	GM12878	IMR90	K562
Downsampled -Best	0.6161	**0.7477**	**0.8984**	0.3260	**0.3196**	**0.4119**	0.3235	0.7727	0.8558	0.4953	0.7384	0.7305
Gaussian -Noise	0.6792	0.7391	0.8842	**0.3271**	0.3066	0.3952	0.2443	0.6653	0.8496	**0.5958**	0.7318	0.7266
Uniform -Noise	0.5939	0.7398	0.8894	0.3234	0.3136	0.3972	0.3357	0.7130	0.8546	0.4667	0.7389	0.7294
Random -Noise	0.6086	0.7394	0.8906	0.3266	0.3138	0.4006	0.3294	0.7142	0.8554	0.4684	0.7388	0.7298
Downsampled -Ensemble	0.4691	0.7337	0.8477	0.2902	0.3061	0.3855	0.4016	**0.7785**	**0.8589**	0.4677	0.7198	**0.7333**
Real-World -LRC	0.6757	0.7170	0.8793	0.2813	0.3025	0.3599	**0.7351**	0.7416	0.8533	0.5904	**0.7475**	0.7235
Real-World-LRC-Ensemble	**0.7151**	0.5387	0.6476	0.3164	0.2483	0.2876	0.6003	0.7478	0.7529	0.5150	0.6868	0.6337

Table S6: This table contains the evaluation results of all the evaluated retraining modalities divided into two tables based on the HiC experiment. HiCNN retrained with the LRC dataset shows the best performance in 11 out of the 24 combinations of datasets and HiC specific similarity metrics.