Generative Augmentation in Sparse Data Regimes: A Controlled Factorial Study in Chinese Character Classification

Joseph Catanzarite

Johns Hopkins University – Whiting School of Engineering

EN.705.603 – Introduction to Generative AI

Spring 2026 – Final Submission

Abstract

Deep learning models for Chinese character recognition face a fundamental challenge: thousands of visually intricate character classes, each requiring labeled examples that are expensive to collect. This paper investigates generative data augmentation — training a generative model on scarce real data and sampling synthetic examples from it — using Chinese-MNIST, a 15-class 64×64 grayscale analogue of the classic MNIST (Modified National Institute of Standards and Technology) handwritten-digit benchmark, as a controlled proxy for low-resource domains such as medical imaging. We train a Conditional Wasserstein Generative Adversarial Network with Gradient Penalty (C-WGAN-GP) and evaluate its synthetic output as augmentation for a Convolutional Neural Network (CNN) classifier. With 50 real images per class, the CNN baseline reaches 88.49% accuracy. Halving the training set to 25 images per class drops accuracy to 71.77% — a loss of 16.7 percentage points. Generative augmentation recovers 7.2 percentage points of those ($p < 0.01$, 4:1 synthetic-to-real ratio), buying back roughly 43% of the accuracy lost to scarcity, at no additional labeling cost. To identify the mechanism, we run a controlled 2×2×2 factorial (real-sample count × critic-filtered versus unfiltered selection × augmentation ratio at 1:1 and 4:1), plus a real-only baseline at each scarcity level, drawing filtered and unfiltered samples as two selection rules over a single critic-scored pool. The gain proves regime-dependent: augmentation helps a data-starved classifier but *degrades* a near-saturated one (50 real images/class, up to −2.2 points, $p < 0.01$). The filter tested here is Stage 1 of a planned two-stage design: it ranks synthetic samples by the trained critic's realism score (critic-based quality filtering, Stage 1). Holding sample count and ratio fixed, Stage 1 filtering produces only a small positive change in downstream accuracy: the four isolation contrasts are all positive but ≤ 0.55 points, and pooled across the four matched conditions the seed-level effect is +0.32 points (95% CI [+0.04, +0.59]; paired $t = 3.21$, $df = 4$, $p = 0.033$; permutation $p = 0.063$) — a small, consistent, borderline-significant positive effect, negligible beside the 6–7-point scarcity gap. Plain augmentation therefore captures nearly all of filtering's benefit. Stage 2 — a perceptual screen based on SSIM (Structural Similarity Index Measure) and LPIPS (Learned Perceptual Image Patch Similarity) distance — was not implemented in this study and may yet show a distinct effect. An earlier diagonal design that paired each scarcity regime with a single selection rule would have misattributed the scarcity gain to filtering; the factorial corrects it. We present this as a sizable, reliable scarcity-regime result and as a cautionary example of how confounded augmentation designs can manufacture a large spurious mechanism out of what is at most a small real one. The three-architecture comparison

proposed at midpoint — spanning variational, adversarial, and hybrid conditional generators, along with adding a Stage 2 filter — are the natural extensions of this study.

1. Introduction

1.1 The Problem: Data Scarcity in a High-Cardinality Domain

The Chinese writing system is, by any measure, one of the most complex visual languages humans have ever produced. The Unicode CJK (Chinese, Japanese, Korean) block alone contains over 20,000 characters, each a unique configuration of strokes layered and balanced according to compositional rules developed over four millennia. A machine learning model needs a large number of labeled training examples per class to recognize these characters reliably. The model used here is a Convolutional Neural Network (CNN) — a deep learning architecture that scans images through layers of learnable filters to detect patterns.

This is the core problem. Collecting and labeling thousands of handwritten examples per character is expensive and slow. When training data is sparse, CNNs tend to *overfit*: rather than learning the underlying geometric logic of a character — its stroke structure, its proportions, its radical components — the model memorizes the specific pixel patterns it has seen and fails to generalize to new examples. The result is a classifier that performs well on its training set but poorly in the real world.

One powerful remedy is **data augmentation**: artificially expanding the training set by creating new examples. Simple augmentation applies geometric transforms to existing images — rotating, flipping, cropping, adding noise. These are useful, but they are bounded by the original data: every augmented image is still a transformation of something that already existed. **Generative data augmentation** is more ambitious. Instead of transforming existing images, we train a generative model — a neural network that learns the underlying distribution of the data — and sample entirely new examples from that learned distribution. Done well, the synthetic images are not copies or distortions of training examples; they are novel, plausible instances that the model has, in a meaningful sense, imagined.

1.2 Three Architectures, One Question

Three families of conditional generative models have emerged as strong candidates for this task, each with a distinct philosophy:

The Conditional Variational Autoencoder (C-VAE) [1, 2] takes a probabilistic approach. It learns to compress images into a structured, low-dimensional *latent space* — a kind of internal coordinate system where similar characters cluster together — and then reconstructs images from points in that space. The "conditional" part means the model is always told which character class it is working with, so generation is class-specific. The VAE is mathematically principled and trains stably, but its outputs can be slightly blurry because it optimizes a pixel-level reconstruction objective.

The Conditional Generative Adversarial Network (C-GAN) [3, 4] takes an adversarial approach. Two networks — a Generator that creates fake images and a Discriminator that tries

to tell fakes from real ones — compete in a minimax game. Over time, the Generator learns to produce images realistic enough to fool the Discriminator. GANs tend to produce sharper, more photorealistic outputs than VAEs, but they are notoriously harder to train: the adversarial balance can tip, leading to *mode collapse* (where the Generator finds a few images that fool the Discriminator and stops exploring) or outright training instability.

The Conditional VAEGAN (C-VAEGAN) [5, 6] is a hybrid that aims to capture the best of both. It uses the VAE's encoder-decoder structure to maintain a coherent latent space, but replaces the VAE's blurry pixel-level reconstruction loss with the GAN's perceptual "does this look real?" criterion. Theoretically, this should yield outputs that are both structurally coherent *and* visually sharp. Whether this advantage holds empirically, on Chinese character data specifically, is one of the central questions this paper investigates.

The midpoint proposal framed all three architectures for implementation on a shared dataset, evaluation on a shared metric suite, and comparison of downstream augmentation utility, under the deliberately open question: *which generative paradigm best serves the augmentation use case for Chinese character recognition?* The final study, as described in Section 1.4, executes the C-GAN arm of that design in depth — realized as the C-WGAN-GP — and the cross-architecture comparison remains the design's natural completion.

1.3 Research Hypotheses

H_1 (Generation Quality): The C-VAEGAN will achieve a lower Fréchet Inception Distance (FID), a higher Structural Similarity Index Measure (SSIM), and better Learned Perceptual Image Patch Similarity (LPIPS) scores than either the C-VAE or C-GAN individually.

H_2 (Augmentation Efficacy): A CNN classifier trained on real data augmented with C-VAEGAN synthetic samples will achieve significantly higher validation accuracy than one trained on real data alone, with $p < 0.05$ on a paired t-test across five independent experimental runs.

H_3 (Quality Filter Value): The novel two-stage quality filter introduced in this paper will improve augmentation efficacy relative to using unfiltered synthetic samples, for all three generative architectures. (As Section 1.4 details, the final study tests H_3 for the C-GAN architecture; the cross-architecture portion is future work.)

1.4 Scope of the Present Study

The midpoint proposal specified a three-architecture comparison (C-VAE, C-GAN, C-VAEGAN). Under compute and schedule constraints, the final study descopes to a deep evaluation of one architecture — the C-GAN trained with the Wasserstein Gradient Penalty objective (C-WGAN-GP) — and tests the role of synthetic-data quality directly. An initial pair of experiments (Section 6.1) appeared to show that critic filtering converts harmful augmentation into beneficial augmentation; a controlled 2×2×2 factorial (Section 6.4) then revealed that this apparent effect was largely a data-scarcity confound, and that Stage 1 critic filtering contributes only a small, borderline-significant improvement over plain augmentation at fixed sample count and ratio. This descoping partially preserves the project's most novel element — the two-stage quality filter proposed in Section 5.4 — by implementing and testing Stage 1 (critic-score ranking) for the C-GAN architecture. Stage 2 (SSIM/LPIPS perceptual screening) remains

planned future work and is not tested here. Hypothesis H_1, which requires the cross-architecture comparison, is explicitly not tested and is carried forward as future work. Hypothesis H_2 is tested for the C-GAN in both filtered and unfiltered regimes; the result — that augmentation's benefit is large under scarcity and is changed only slightly, by a small borderline-significant positive increment, under Stage 1 filtering — is the paper's central finding.

2. Related Work

2.1 Variational Autoencoders

The Variational Autoencoder (VAE), introduced by Kingma and Welling in 2014 [1], was a landmark contribution to generative modeling. At its core, a VAE is an encoder-decoder architecture: the encoder maps an input image to a probability distribution in a low-dimensional latent space, and the decoder reconstructs the image from a sample drawn from that distribution. The key insight is that by forcing the latent distribution to remain close to a standard normal distribution, the model learns a smooth, continuous latent space in which nearby points decode to visually similar images — a structure that makes controlled, interpolated generation possible.

Sohn, Lee, and Yan [2] extended this framework with conditioning, producing the Conditional VAE (C-VAE). By feeding the class label into both the encoder and decoder, the C-VAE can generate examples of a specific class on demand — exactly what is needed for a labeled augmentation pipeline.

2.2 Generative Adversarial Networks

Goodfellow and colleagues introduced Generative Adversarial Networks (GANs) in 2014 [3] with a deceptively simple idea: train two networks simultaneously. A Generator network G learns to map random noise vectors to realistic-looking images; a Discriminator network D learns to distinguish real images from G's fakes. Each improves by trying to beat the other. When training converges, G has learned to produce images indistinguishable from the real data distribution. Mirza and Osindero [4] introduced the conditional extension (C-GAN) by providing the class label as an additional input to both G and D, enabling class-specific generation. Arjovsky and colleagues [13] later reframed adversarial training around the Wasserstein distance (the Wasserstein GAN, or WGAN), and Gulrajani and colleagues [7] stabilized it further with the Gradient Penalty technique (WGAN-GP), which dramatically improves GAN training by penalizing the Discriminator for violating a smoothness constraint — addressing one of the most common failure modes in adversarial training.

2.3 VAE-GAN Hybrids

Larsen and colleagues [5] proposed combining the two paradigms: use the VAE's encoder-decoder as the generative backbone, but replace the VAE's pixel-level reconstruction loss with a learned perceptual loss computed from the GAN Discriminator's internal feature representations. The intuition is that pixel-level loss penalizes the model for every slightly misplaced pixel — leading to blurriness as a conservative hedge — whereas a perceptual loss

asks only whether the image looks right overall. The conditional variant (C-VAEGAN) is directly applicable to Chinese character synthesis, where we need both structural regularity (favoring the VAE's latent structure) and visual sharpness (favoring the GAN's perceptual criterion).

2.4 Chinese Character Recognition and Synthesis

Deep CNN-based recognition of Chinese handwriting has advanced substantially, with state-of-the-art systems achieving over 95% accuracy on benchmark datasets [8]. However, these results depend on very large labeled datasets — the CASIA-HWDB (Center for Analysis and Statistics of Handwriting – Handwritten Database) corpus, for instance, contains nearly 3.9 million character images across 7,356 classes [9]. The Chinese-MNIST dataset [10], a more compact benchmark with 15 character classes and approximately 1,000 images each, provides a tractable starting point for generative augmentation experiments.

Kong and Xu [6] directly addressed Chinese character synthesis using a C-VAEGAN architecture, demonstrating feasibility on a small 200-samples-per-class regime. Their work encountered training instability — discriminator loss collapsing to near zero, and KL (Kullback–Leibler) divergence numerical instability after approximately eight training epochs — and reported no downstream augmentation evaluation. This paper extends their work in three directions: a systematic three-model comparison, a downstream classification evaluation, and the introduction of a quality filtering pipeline.

2.5 Positioning of This Work

To the author's knowledge, no prior published work has conducted a controlled three-way comparison of C-VAE, C-GAN, and C-VAEGAN specifically for Chinese character data augmentation with downstream OCR (Optical Character Recognition) accuracy as the evaluation criterion. This comparison directly addresses a gap in the literature: not which model generates the prettiest images, but which generative paradigm most usefully serves a downstream classification task.

3. Research Problem Statement

We begin with a labeled training set $D_{real} = \{(x_i, y_i)\}$ of Chinese character images, where each x_i is a grayscale image and each y_i is one of C character class labels. Our generative model — in the executed study, the C-WGAN-GP — is trained on this real data and used to produce a synthetic dataset D_{synth}. We then train a CNN classifier on the augmented set $D_{real} \cup D_{synth}$ and compare its performance to a baseline CNN trained on D_{real} alone. The central question is:

> *As proposed at midpoint: which conditional generative architecture — C-VAE, C-GAN, or C-VAEGAN — produces synthetic data that most effectively augments the real training set for downstream Chinese character classification? As executed in the final study: does C-WGAN-GP synthetic data augment that training set effectively, and does critic-based quality filtering change the answer?*

This decomposes into three measurable sub-questions: (1) Which architecture produces the highest-quality synthetic characters, as measured by FID, SSIM, and LPIPS? (2) Which produces

the greatest lift in CNN classifier accuracy when added to the real training data? (3) Does the novel two-stage quality filter improve augmentation efficacy, and does its benefit differ across architectures?

Primary dataset: Chinese-MNIST — 15 numeral-character classes: the ten digits zero through nine (*零*、*一*、*二*、*三*、*四*、*五*、*六*、*七*、*八*、*九*) plus the five magnitude characters for ten, hundred, thousand, ten thousand, and one hundred million (*十*、*百*、*千*、*万*、*亿*); 1,000 grayscale 64×64-pixel images per class, 15,000 images total.

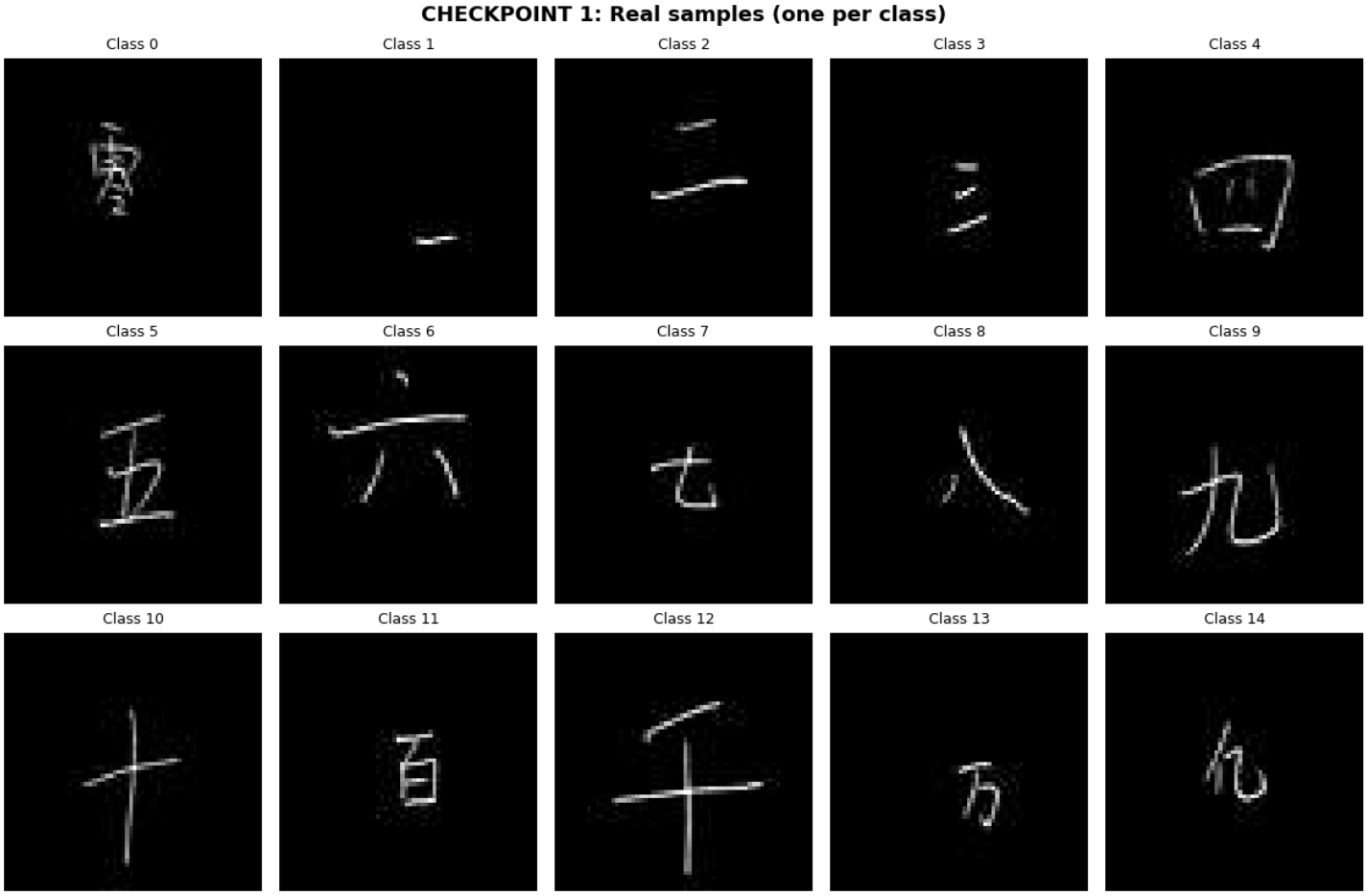

Figure 2 — Real training samples from Chinese-MNIST: one image per class at 64×64 pixels. Top row: numerals zero through four (零、一、二、三、四); middle row: numerals five through nine (五、六、七、八、九); bottom row: magnitude characters ten, hundred, thousand, ten thousand, one hundred million (十、百、千、万、亿). Drawn from the real training split; images illustrate the stroke complexity that motivates generative augmentation under scarcity.

Performance criteria: Generation quality (FID, SSIM, LPIPS), downstream CNN classification accuracy, and paired t-test statistical significance ($\alpha = 0.05$) across five independent runs.

4. Evaluation Metrics

Before describing the architectures, it is worth pausing to explain precisely how we will measure success. A generative model can fail in many ways: its images might look nothing like real characters, they might all look like the same character regardless of the conditioning label, or they might be technically realistic but somehow not useful for training a classifier. Each of our three metrics captures a different dimension of this question.

4.1 Fréchet Inception Distance (FID)

The Fréchet Inception Distance, or FID, asks: how similar is the *distribution* of generated images to the distribution of real images? It is the most widely used metric for evaluating generative models, and for good reason — a model that produces high-quality individual images but misses whole regions of the real distribution (mode collapse) will score poorly on FID even if individual samples look good.

To compute FID, both sets of images — real and generated — are passed through a pretrained Inception-v3 network (a deep CNN trained on ImageNet), and the activations from one of its intermediate layers are extracted. These activations form a high-dimensional feature space in which images with similar visual characteristics cluster together. We model each set's feature activations as a multivariate Gaussian distribution, characterized by a mean vector μ and covariance matrix Σ. The FID is then the Fréchet distance between these two Gaussians:

$$FID = ||\mu_r - \mu\hat{}||^2 + Tr(\Sigma_r + \Sigma\hat{} - 2(\Sigma_r\Sigma\hat{})^{(1/2)})$$

Here, μ_r and Σ_r are the mean and covariance of the real image features, and $\mu\hat{}$ and $\Sigma\hat{}$ are those of the generated image features. $Tr(\cdot)$ is the matrix trace. The term $||\mu_r - \mu\hat{}||^2$ measures how far apart the centers of the two distributions are; the trace term measures how well their shapes match. A lower FID is better — a perfect generative model that exactly reproduces the real distribution would score zero. In practice, FID scores below 10 are considered excellent; scores above 50 suggest significant quality problems.

4.2 Structural Similarity Index Measure (SSIM)

While FID measures distributional similarity at the population level, the Structural Similarity Index Measure (SSIM) measures the similarity between two individual images. Introduced by Wang and colleagues [11], SSIM is motivated by a simple observation: the human visual system is particularly sensitive to structural information — the spatial arrangement of luminance patterns — rather than absolute pixel values. A metric that penalizes every slightly misaligned pixel equally (like mean squared error) misses this point.

SSIM decomposes image similarity into three components — luminance (l), contrast (c), and structure (s) — and combines them multiplicatively:

$$SSIM(x, y) = l(x,y) \cdot c(x,y) \cdot s(x,y)$$

where each component is computed over local image patches. The luminance term $l(x,y) = (2\mu_x\mu_y + C_1) / (\mu_x^2 + \mu_y^2 + C_1)$ compares mean pixel values; the contrast term $c(x,y) = (2\sigma_x\sigma_y + C_2) / (\sigma_x^2 + \sigma_y^2 + C_2)$ compares standard deviations; and the structure term $s(x,y) = (\sigma_{xy} +$

C_3) / ($\sigma_x\sigma_y + C_3$) compares the normalized cross-correlation between patches. C_1, C_2, C_3 are small stabilization constants. The result is a value in [-1, 1] where 1 means identical images, 0 means no structural correlation, and negative values indicate structural anti-correlation. For Chinese character generation, an SSIM above ~0.6 between a generated image and its nearest real-class neighbor indicates the generated character preserves the essential stroke structure.

4.3 Learned Perceptual Image Patch Similarity (LPIPS)

SSIM is a hand-crafted metric; Learned Perceptual Image Patch Similarity (LPIPS), introduced by Zhang and colleagues [12], asks instead: what does a deep neural network think about the similarity between two images? The idea is that a network trained on large amounts of image data has learned feature representations that correlate with human perceptual judgment far better than any hand-designed formula.

To compute LPIPS, both images are passed through a pretrained deep network (such as VGG — the Visual Geometry Group network — or AlexNet). The activations at multiple layers are extracted, normalized, and compared channel-by-channel. The differences are then aggregated into a single scalar distance:

$$LPIPS(x, y) = \Sigma_l \, w_l \, ||\varphi_l(x) - \varphi_l(y)||^2$$

where $\varphi_l(\cdot)$ denotes the feature activations at layer l, and w_l are learned weights that calibrate the contribution of each layer. Unlike SSIM, LPIPS is a *distance* — lower is better, meaning the two images are perceptually more similar. A generated Chinese character with a low LPIPS distance to a real character of the same class not only has the right pixel structure, it looks right to a network that has learned something like visual perception.

4.4 Downstream CNN Classification Accuracy

All three of the above metrics evaluate the quality of generated images in isolation. But the ultimate test for our purposes is pragmatic: does adding the synthetic data actually help a classifier learn? We measure this by training a CNN classifier under two conditions — on real data alone (the baseline) and on real plus synthetic data (the augmented condition) — and comparing their validation accuracy. We repeat each training run five times using the same set of random seeds across all conditions. This serves three purposes. First, multiple runs reveal whether an accuracy difference is consistent or a product of a lucky initialization. Second, the standard deviation across the five runs provides an uncertainty estimate for each reported accuracy — this is what the Std column in the results tables and the error bars in Figure 5 reflect. Third, because every condition shares the same seeds, each seed constitutes a matched pair — baseline seed 42 versus augmented seed 42, and so on — allowing a paired t-test that cancels seed-level variance and isolates the effect of the treatment (augmentation, filtering, or ratio) from random training noise. Statistical significance is assessed at $p < 0.05$. We also compute Cohen's d, a measure of effect size that tells us not just whether the difference is statistically significant, but how large it is in practical terms.

5. Method

5.1 Experimental Pipeline Overview

The experimental scaffold was designed to be architecture-agnostic, so that all three proposed architectures could be evaluated under identical conditions; the final study instantiates it for the C-WGAN-GP arm. (Two abbreviations appearing in the comparison table below — ELBO, the Evidence Lower Bound, and GP, the Gradient Penalty — are developed fully in Section 5.2.) The pipeline has four stages: (1) train the generative model on the real training set; (2) generate a synthetic dataset; (3) optionally filter the synthetic dataset through the quality pipeline; (4) train the CNN classifier on the combined real-plus-synthetic set and evaluate on held-out real data. Everything outside the synthetic-data path is held constant. Any observed difference in downstream accuracy is therefore attributable to the synthetic data and its curation — in the executed experiments, to the presence and filtering of C-WGAN-GP samples — and not to any other experimental variable.

Figure 1 presents the experimental pipeline as a block diagram: real data is split once (stratified, seeded); the C-WGAN-GP trains on the real training subset; the trained Generator produces a large candidate pool which the trained Critic scores; the quality filter retains the top-scoring samples; and the CNN classifier is trained on real-plus-filtered-synthetic data and evaluated on the untouched real test set, with a real-only baseline path for comparison.

Figure 1 — Experimental pipeline: C-WGAN-GP generation, critic-based quality filtering, and downstream classification evaluation

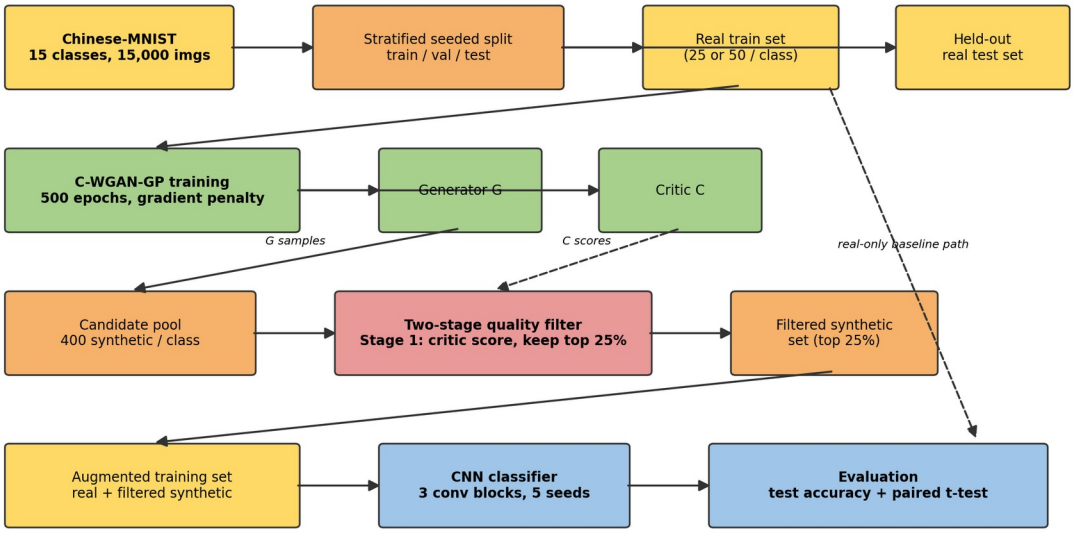

Figure 1 — Experimental pipeline block diagram. Color coding: gold boxes are the datasets used for training and evaluation (real and augmented); green boxes are the generative components (C-WGAN-GP training, Generator, Critic); the red box is the two-stage quality filter, this paper's original contribution; blue boxes are the evaluation components (CNN classifier and statistical testing); orange boxes are intermediate pipeline elements — the data split operation and the synthetic-data pools flowing into and out of the filter. Dashed arrows mark the critic-scoring path and the real-only baseline path.

Property	C-VAE	C-GAN	C-VAEGAN (Hybrid)
Training objective	ELBO (reconstruction + β·KL divergence)	Minimax adversarial game	ELBO + adversarial perceptual loss
Latent space	Structured, continuous N(0,I) prior	Unstructured noise vector z	Structured (VAE encoder) + GAN perceptual loss
Output quality	Smooth; may be slightly blurry	Sharp; risk of mode collapse	Sharp and structured (theoretical best)
Training stability	High — ELBO is well-defined	Medium — collapse risk without GP	Medium-High — GP regularization required
Key citations	[1], [2]	[3], [4], [7]	[5], [6]

5.2 Mathematical Foundations of Each Architecture

The C-VAE and the Evidence Lower Bound (ELBO)

The C-VAE rests on a beautiful probabilistic idea: rather than learning a fixed encoding for each image, learn a *distribution* over possible encodings. Formally, given an image x and its class label y, the encoder produces the parameters (μ, σ) of a Gaussian distribution $q\varphi(z \mid x, y)$ over a latent vector z. The decoder then takes a sample z from this distribution (along with y) and reconstructs the image. The model is trained to maximize the Evidence Lower Bound (ELBO), which balances two competing objectives:

$$ELBO = E_{q\varphi(z|x,y)} [\log p\theta(x \mid z, y)] - \beta \cdot KL(q\varphi(z \mid x, y) \,||\, p(z))$$

The first term is the **reconstruction loss**: how well does the decoder reconstruct the original image from the sampled z? The second term is the **KL (Kullback–Leibler) divergence**: how far is the learned encoding distribution $q\varphi(z \mid x, y)$ from a standard normal distribution $p(z) = N(0, I)$? The KL term acts as a regularizer that keeps the latent space organized — preventing the model from using arbitrary, scattered encodings. The hyperparameter β controls the trade-off: higher β produces a smoother, more regular latent space at some cost to reconstruction sharpness.

The Reparameterization Trick

There is a subtle but critical engineering challenge lurking in the ELBO: to optimize the model with gradient descent, we need to backpropagate gradients through the *sampling step* $z \sim q\varphi(z \mid x, y)$. But sampling is a stochastic operation, and stochastic operations are not differentiable in the usual sense. Gradients cannot flow through randomness.

The **reparameterization trick** is the elegant solution. Instead of sampling z directly from the learned distribution $N(\mu, \sigma^2)$, we rewrite the sampling operation as:

$$z = \mu + \sigma \odot \varepsilon, \varepsilon \sim N(0, I)$$

Here, ε is a sample from a *fixed* standard normal distribution — it carries all the randomness — while μ and σ are deterministic outputs of the encoder network. Think of it this way: imagine you need to draw a random point from a circle of radius r centered at point p. You

could sample directly, but gradients would not flow. Instead, you sample a random direction ε from a unit circle, then *compute* the point as p + r·ε. The randomness lives in ε, which we treat as a fixed input, not a parameter; all the learnable structure lives in p and r, through which gradients flow freely. The reparameterization trick shifts the randomness out of the computational graph and into a fixed noise term, making the entire operation differentiable with respect to μ and σ.

The C-GAN and Gradient Penalty

The C-GAN trains a Generator G and Discriminator D simultaneously with opposing objectives. Conditioned on class label y, G maps a noise vector z to a synthetic image G(z | y); D takes an image (real or fake) and its label y and outputs a probability that the image is real. The minimax objective is:

$$min\_G\ max\_D\ [\ E[log\ D(x \mid y)] + E[log(1 - D(G(z \mid y)))]\ ]$$

The Generator tries to minimize this (fool D); the Discriminator tries to maximize it (detect fakes). To stabilize training, we apply the Wasserstein Gradient Penalty (WGAN-GP) [7], which adds a term penalizing the Discriminator for having gradients too far from unit norm:

$$L^p \mathrel{+}= \lambda \cdot E[(||\nabla^p D(\hat{x} \mid y)||\_2 - 1)\hat{}2]$$

where $\hat{x}$ is a random convex combination of real and generated images. This constraint keeps the Discriminator smooth, preventing the gradient signals sent to the Generator from exploding or vanishing — the root cause of most GAN training instability.

The C-VAEGAN Combined Objective

The C-VAEGAN uses the VAE encoder-decoder as its generative backbone, but replaces the VAE's pixel-level reconstruction loss with a perceptual loss computed from the GAN Discriminator's internal feature representations. The combined loss is:

$$L = L^{OMLN} + \lambda_a{}^{dv} \cdot L^{NAN} + \lambda^{feat} \cdot L^{feat}{}_r{}^m$$

The ELBO term maintains the structured latent space; the adversarial term sharpens outputs by asking the Discriminator whether they look real; the feature-matching term aligns the internal representations of real and reconstructed images within the Discriminator's layers, further improving perceptual quality.

5.3 Shared Architecture

To ensure a fair comparison, the design specifies a single convolutional backbone shared by all three proposed models; the C-WGAN-GP trained in the final experiments instantiates it. The encoder (used in C-VAE and C-VAEGAN) consists of four convolutional layers with Batch Normalization and LeakyReLU activations, reducing the input from 64×64 to a 4×4 feature map, then projecting to the latent parameters (μ, log σ^2). The class label is concatenated to the image as a tiled channel before the first convolutional layer. The decoder/generator uses four transposed convolutional layers (the "reverse" of convolution, used to upscale feature maps back to image size), producing a 64×64 grayscale output through a Tanh activation that maps

values to [-1, 1]. The discriminator uses the same four-layer convolutional structure as the encoder, ending in a scalar logit.

5.4 The Two-Stage Quality Filter (Original Contribution)

A critical and independently developed contribution of this paper is the two-stage quality filter, designed to curate synthetic samples before they enter the augmentation pool. The motivation is straightforward: generative models — particularly early in training or in underrepresented character classes — do not always produce good samples. Adding low-quality synthetic images to the training set could harm the classifier rather than help it. Filtering before augmentation ensures that only synthetic samples meeting minimum quality standards are used.

To the author's knowledge, this two-stage filter design — combining discriminator confidence with perceptual distance as a principled quality gate for generative augmentation — has not been previously reported in the literature for the Chinese character domain. It was developed independently in the course of this project.

Stage 1 — Discriminator confidence: A synthetic image $\tilde{x}$ is accepted only if the trained Discriminator assigns it a confidence score above a threshold τ_D (approximately the top 70th percentile of generated samples). This stage filters out samples that the model itself "knows" are poor — images that look so far from the real distribution that even the Discriminator is not fooled.

Stage 2 — Perceptual proximity: Among samples passing Stage 1, we additionally require that the SSIM score between $\tilde{x}$ and its nearest real-class neighbor exceeds τ_S, and that the LPIPS distance is below τ_L. This stage filters out samples that may score well on the Discriminator but are perceptually dissimilar from any real example in the class — stylistically plausible but geometrically wrong. The thresholds are tuned on a held-out validation split.

The filter is designed to apply identically across architectures, enabling a clean A/B comparison between filtered and unfiltered augmentation. In the final study this comparison is realized for the C-GAN architecture, providing the test of H_3 reported in Section 6.

Implementation in the final study. Stage 1 was implemented and tested under a controlled design. For each class, a single pool of 800 candidate images is generated from the trained generator and scored once by the trained WGAN critic (conditioned on the class label). Two selection rules are then applied to that same pool: the filtered condition takes the n highest-scoring images per class — a strict top-k by critic score — and the unfiltered condition takes a random n per class. Because n is the per-condition synthetic count, the filtered slice is the strictest available: it ranges from the top 3.1% of the pool (1:1 ratio, 25 real images/class) to the top 25% (4:1 ratio, 50 real images/class). It is therefore neither a uniform quartile nor a random draw from one. Because the Wasserstein critic outputs an unbounded realism score rather than a probability, the threshold τ_D is realized as a per-class rank cutoff rather than an absolute confidence value — a natural adaptation that requires no calibration. Stage 2 (SSIM/LPIPS perceptual screening) was not implemented in the final experiments and is carried forward as future work; the results of Section 6 therefore test Stage 1 alone.

5.5 CNN Classifier

The evaluation classifier is a standard three-block CNN: each block consists of two Conv2d layers with Batch Normalization and ReLU activations followed by MaxPooling, reducing spatial dimensions from 64×64 down to an 8×8 feature map. Two fully connected layers with Dropout ($p = 0.4$) and a softmax output over 15 classes complete the architecture. The same classifier is used in all ten conditions: the two real-only baselines (25 and 50 real images/class) and every augmented variant (unfiltered and critic-filtered, at 1:1 and 4:1 ratios, at both scarcity levels). All are trained with the Adam optimizer and a cosine learning rate schedule. Holding the classifier fixed ensures that differences in accuracy between conditions reflect the quality of the augmentation, not the classifier design.

5.6 Hyperparameters Used

The midpoint proposal specified a hyperparameter search across latent dimension, KL weight, gradient-penalty coefficient, and augmentation ratio. Under the descoped final design, fixed literature-standard values were used for the generative model, and the augmentation-ratio dimension of the search was retained (1:1 and 4:1 synthetic-to-real evaluated in both experiments). The fixed values follow the WGAN-GP reference settings of Gulrajani et al. [7]:

Parameter	Value	Rationale
Latent dimension d	128	Upper midpoint candidate; capacity for 15-class conditional generation
Gradient penalty λ	10	WGAN-GP reference default [7]
Critic steps per generator step	5	WGAN-GP reference default [7]
Optimizer (G and C)	Adam, lr 1e-4, $\beta_1 = 0.0$, $\beta_2 = 0.9$	WGAN-GP reference settings [7]
WGAN-GP training epochs	500	Selected after a 100-epoch pilot showed non-converged generator loss
CNN: epochs / lr / dropout	30 / 1e-3 (cosine) / 0.4	Fixed across all conditions for comparison fairness
Augmentation ratio (searched)	1:1, 4:1	Evaluated in both experiments
Filter pool / retention	800/class pool; top-k by critic score	Stage 1 quality filter operating point (not tuned)

A full search over the generative hyperparameters and the filter retention threshold is deferred to future work; Section 7 discusses the threshold sweep as a priority follow-up.

6. Experimental Results

The study proceeds in two stages. An initial pair of experiments (Section 6.1) follows a diagonal design — unfiltered augmentation at mild scarcity, critic-filtered augmentation at deep scarcity — and appears to show that filtering converts harmful augmentation into beneficial augmentation. Because that design varies filtering and scarcity together, a controlled 2×2×2 factorial (Section 6.4) then disentangles them. All experiments share the identical stratified split, five random seeds (42–46), CNN classifier, and held-out real test set (100 images/class, 1,500 total). All significance tests are paired t-tests across seeds against the within-regime baseline, $\alpha = 0.05$.

Experiment 1 — unfiltered augmentation under mild scarcity. The C-WGAN-GP was trained for 500 epochs on 50 real images/class (750 total). Synthetic images were sampled without curation and mixed with the real training data at 1:1 and 4:1 synthetic-to-real ratios.

Experiment 2 — critic-filtered augmentation under deep scarcity. The real training set was reduced to 25 images/class (a strict subset of Experiment 1's training set, leaving val/test untouched). Synthetic data was drawn from the same 500-epoch generator checkpoint, but passed through the Stage 1 quality filter as originally implemented: 400 candidates/class scored by the trained critic, top 25% retained. (The controlled factorial of Section 6.4 standardizes this on the larger 800/class pool and the strict top-k selection of Section 5.4.)

6.1 Initial Experiments (Diagonal Design)

Experiment	Condition	Mean Acc	Std	Δ vs Baseline	p-value	Significant?
Exp 1 (50/class, unfiltered)	Baseline	88.49%	0.78%	—	—	—
	Augmented 1:1	87.44%	0.89%	−1.05%	0.057	No
	Augmented 4:1	86.27%	0.25%	−2.22%	0.008	**Yes (−)**
Exp 2 (25/class, filtered)	Baseline	71.77%	2.57%	—	—	—
	Augmented 1:1	77.32%	1.10%	+5.55%	0.0085	**Yes (+)**
	Augmented 4:1	80.19%	1.63%	+8.42%	0.0005	**Yes (+)**

Read at face value, these two experiments suggest a clean story. Unfiltered synthetic data degrades accuracy, with the degradation growing with ratio. Critic-filtered synthetic data improves it, with the improvement also growing with ratio — every individual seed improved in both filtered conditions. The filtered 4:1 augmentation condition recovers nearly half the accuracy lost to halving the real data (71.77% → 80.19%, against an 88.49% baseline at 50 real images/class). This is the reading reported in earlier versions of this work. It is also

confounded: the harmful condition (unfiltered) sits at 50 real images/class and the beneficial condition (filtered) at 25, so filtering and scarcity vary together, and the sign reversal cannot be attributed to filtering alone. Section 6.4 removes the confound with a full factorial.

6.2 Training Dynamics

The C-WGAN-GP exhibited the loss behavior characteristic of a healthy gradient-penalty run: large early oscillations (epochs 1–20), followed by stabilization, with the critic loss settling near −4.8 and the generator loss plateauing in the 6.0–6.8 band from roughly epoch 200 onward. A 100-epoch pilot run, by contrast, ended with the generator loss still rising — and synthetic samples from that checkpoint, when used for augmentation, produced a 10.96-percentage-point accuracy *drop* (77.05% vs the 88.01% baseline measured in the pilot configuration). Extending training to 500 epochs reduced unfiltered augmentation's harm to the −1% to −2% range reported above. Generator convergence is thus necessary for augmentation to be even modestly safe; the controlled factorial of Section 6.4 shows that what then makes augmentation beneficial is overwhelmingly data scarcity, with Stage 1 quality filtering adding only a small increment.

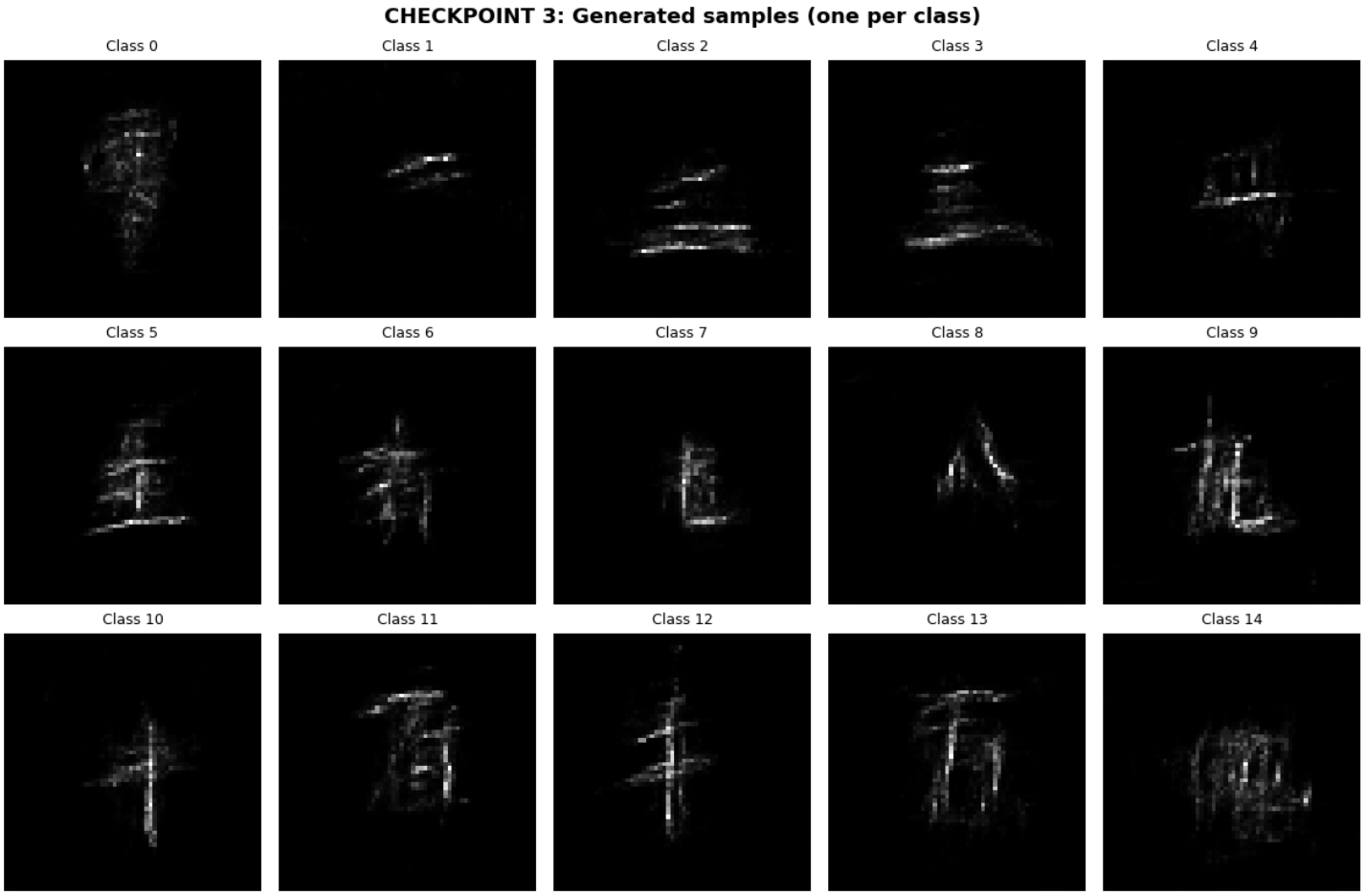

Figure 3 — Synthetic characters produced by the C-WGAN-GP after 500 training epochs: one generated sample per class, arranged in the same class order as Figure 2. Visual quality is sufficient for downstream augmentation use; stroke structure is broadly recognizable, though

fine details vary by class. These samples are drawn from the same generator checkpoint used in all factorial conditions.

Classifier training loss provides the mechanism's signature. In the unfiltered 4:1 condition, final-epoch training loss fell to 0.01–0.02 (versus 0.09–0.28 for the baseline) with the lowest seed variance of any condition — the classifier reliably memorized the synthetic distribution's artifacts rather than learning transferable character structure. In the filtered conditions, training loss remained in a healthy intermediate range and test variance *fell* relative to the scarce baseline (1.10% vs 2.57% at 1:1), indicating the filtered synthetic data acted as a stabilizer as well as a signal source.

6.3 Quality Filter Analysis

Visual inspection across the filter boundary confirms the critic ranks usefully. Retained (top-scoring) samples show cleaner stroke structure and fewer brightness artifacts. Low-scoring samples include the spatially discontinuous and low-contrast failures characteristic of the generator's weaker outputs. The critic therefore separates visibly cleaner samples from weaker ones. The factorial of Section 6.4 shows, however, that this visible ranking advantage translates into only a small downstream accuracy gain: at fixed real-sample count and ratio, classifiers trained on the top-scoring samples beat those trained on a random draw from the same pool by a slim margin — individually within noise, but consistently positive and, pooled across conditions, borderline-significant (Section 6.4). The relationship between retention threshold and downstream benefit is otherwise unmeasured and remains a priority follow-up. Figure 4 illustrates the visual distinction between the two selection rules.

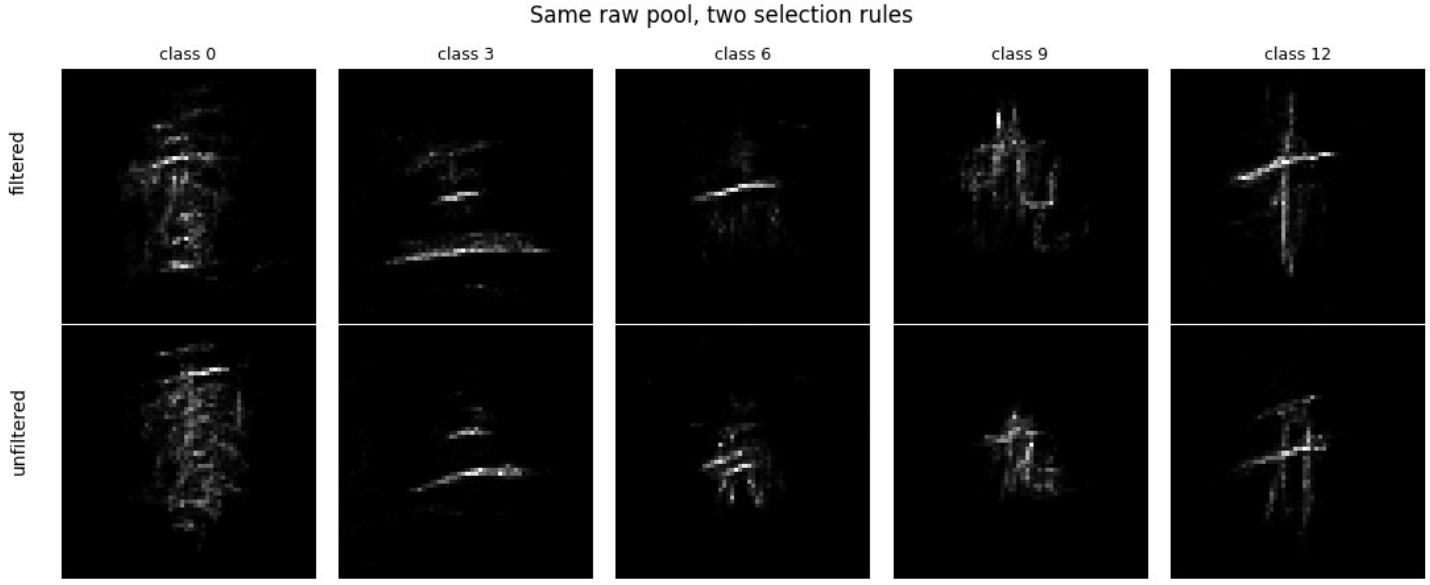

Figure 4 — Filtered (top row) versus unfiltered (bottom row) synthetic samples drawn from the same 800-image critic-scored pool, for five representative classes (classes 0, 3, 6, 9, and 12 of 15; shown for display purposes only — all 15 classes were used in the experiment). Top-scoring samples retained by Stage 1 filtering show cleaner stroke structure and fewer artifacts; low-scoring samples in the unfiltered draw include spatial discontinuities and low-contrast failures.

The critic visibly separates cleaner samples from weaker ones — yet as Section 6.4 shows, this ranking advantage yields only a small, borderline-significant downstream accuracy gain (+0.32 points pooled across the matched conditions) at the operating points tested.

6.4 Controlled Factorial: Scarcity, Not Filtering

The diagonal design confounds filtering with scarcity. To separate them, we run a full factorial over three factors — real-sample count (25 or 50 images/class), selection rule (critic-filtered top-k versus unfiltered random-k), and augmentation ratio (1:1 or 4:1) — plus a real-only baseline at each scarcity level, for ten conditions in all. Filtered and unfiltered samples are drawn from the same 800/class critic-scored pool (Section 5.4), so the only thing that changes between a filtered and an unfiltered cell is whether the n samples are the top-scoring ones or a random draw. This is an independent controlled re-run. Both baselines reproduce the initial experiments exactly (88.49% at 50/class, 71.77% at 25/class), anchoring reproducibility. The synthetic-dependent cells differ from Section 6.1 within noise.

Real/class	Condition	Mean Acc	−/+ vs base	p-value
50	Baseline	88.49%	—	—
50	Unfiltered 1:1	87.05%	−1.44	0.049
50	Unfiltered 4:1	86.35%	−2.15	0.009
50	Filtered 1:1	87.60%	−0.89	0.222
50	Filtered 4:1	86.48%	−2.01	0.001
25	Baseline	71.77%	—	—
25	Unfiltered 1:1	77.52%	+5.75	0.005
25	Unfiltered 4:1	78.79%	+7.01	0.003
25	Filtered 1:1	77.88%	+6.11	0.007
25	Filtered 4:1	79.01%	+7.24	0.004

Figure 5 plots all ten conditions.

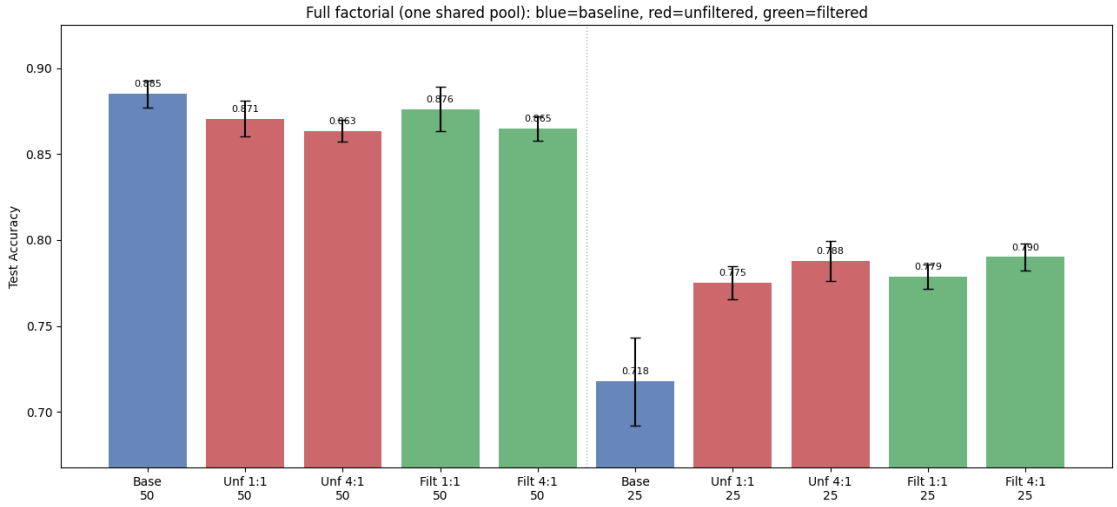

Figure 5 — Classification accuracy across all ten factorial conditions: two scarcity levels (25 and 50 real images/class) × selection rule (unfiltered / critic-filtered) × ratio (1:1 / 4:1), with the two real-only baselines. Blue = baseline, red = unfiltered, green = critic-filtered; the dotted divider separates the 50/class block (left) from the 25/class block (right). Bar labels are mean held-out test accuracy; error bars span ±1 standard deviation across five seeds.

The pattern is unambiguous and it is overwhelmingly about scarcity, not filtering. At 50 real images/class — where the baseline is already near-saturated at 88.49% — every augmented condition degrades accuracy, filtered or not. At 25 real images/class, every augmented condition improves it, filtered or not, by +5.8 to +7.2 points, with the gain growing in the ratio. The best condition recovers about 43% of the 16.7-point accuracy gap that halving the real data opened (the +7.24-point lift at 25/class, 4:1, against a 71.77% baseline). Whether augmentation helps or hurts is set by the data regime; the selection rule barely moves the result.

Isolating the filter

The filter effect is the filtered-minus-unfiltered difference at fixed count and ratio — the four contrasts below. Each holds everything constant except whether the n samples are top-scoring or random, so each is a clean estimate of what critic filtering buys.

Contrast (count, ratio)	Filtered – Unfiltered (pp)	p-value
50/class, 1:1	+0.55	0.034
50/class, 4:1	+0.13	0.78
25/class, 1:1	+0.36	0.55
25/class, 4:1	+0.23	0.64

Every contrast is small — at most 0.55 points, against scarcity effects 10× to 13× larger — but all four are positive. Taken one at a time they are underpowered: three of the four are non-significant, and the one nominally significant contrast (50/class, 1:1, $p = 0.034$) does not survive multiple-comparison correction (with four contrasts the Holm—Bonferroni threshold for the smallest p-value is $0.05/4 = 0.0125$, which 0.034 fails). Per-contrast, then, the effect is hard to see. Pooling recovers it: averaging the filtered-minus-unfiltered difference across the four

matched conditions for each of the five seeds, and testing those five paired differences against zero, gives a mean of +0.32 points (95% CI [+0.04, +0.59]; paired $t = 3.21$, $df = 4$, $p = 0.033$; a permutation/Wilcoxon test gives $p = 0.063$). The honest statement is therefore not that filtering does nothing, but that Stage 1 critic filtering produces a small, consistent, borderline-significant positive change in downstream accuracy once sample count and ratio are held fixed — real, yet negligible beside the 6–7-point scarcity gap, and demonstrated here only for this single generator and candidate pool. This revises the stronger no-detectable-effect reading of earlier versions: the per-contrast tests were simply underpowered to see a sub-half-point effect that pooling makes visible. The critic ranks images by visible quality (Section 6.3), and that ranking does transfer to classifier accuracy — but only weakly. Plain augmentation captures nearly all of the filtered result under every tested condition. Whether Stage 2 of the filter (SSIM/LPIPS perceptual screening, not implemented here) produces a larger effect is the natural next experiment — see Section 7 future work. The diagonal design of Section 6.1 would have over-credited this small filter increment as the augmentation gain's main driver, which it is not; the factorial shows the gain is overwhelmingly a scarcity effect.

6.5 Status of the Designed Metric Suite

The generation-quality metrics specified in Section 4 (FID, SSIM, LPIPS) were not computed in the final experiments; downstream classification accuracy — the metric this paper argues is the decision-relevant one for the augmentation use case — served as the sole quantitative criterion. Computing the full metric suite for retained versus rejected filter samples would link the filter's behavior to measurable image statistics and is listed in future work.

7. Conclusions

This study set out to determine whether conditional generative models can usefully augment scarce training data for Chinese character classification, with the question sharpened at midpoint into three hypotheses. The verdicts:

H_1 (Generation Quality, cross-architecture): *Not tested.* The three-architecture comparison was descoped; H_1 requires the C-VAE and C-VAEGAN implementations and the FID/SSIM/LPIPS metric suite, all carried forward as future work.

H_2 (Augmentation Efficacy): *Supported under scarcity, and regime-dependent.* For the C-WGAN-GP, the answer depends entirely on the data regime. At 50 real images/class the baseline is near-saturated and augmentation degrades accuracy (down to −2.15%, $p = 0.009$, unfiltered 4:1). At 25 real images/class augmentation satisfies H_2 decisively (up to +7.24%, $p = 0.004$, 4:1), improving monotonically with ratio and recovering roughly 43% of the accuracy lost to scarcity. Crucially, the controlled factorial shows this benefit holds whether or not the synthetic samples are critic-filtered.

H_3 (Quality Filter Value): *Weakly supported under controlled test — a small, borderline-significant positive effect.* The initial diagonal experiments appeared to support H_3 strongly, but the controlled factorial cuts that to a sliver for Stage 1: holding sample count and ratio fixed, all four filtered-minus-unfiltered contrasts are positive but ≤ 0.55 points and individually underpowered (three non-significant, the fourth failing Holm–Bonferroni correction). Pooled

across the four matched conditions, however, the seed-level effect is +0.32 points (95% CI [+0.04, +0.59]; paired $t = 3.21$, $df = 4$, $p = 0.033$; permutation $p = 0.063$) — small, consistent, and borderline-significant. The trained critic ranks images by visible quality, and that ranking does transfer into a classifier benefit, but a negligible one beside the scarcity gap and demonstrated only for this single generator and pool. The defensible verdict is thus a small, real Stage 1 filter increment — neither the strong effect H_3 envisioned nor the zero effect earlier versions reported. Stage 2 of the filter (SSIM/LPIPS perceptual screening) was not implemented and remains the priority next test of whether quality filtering can produce a practically meaningful gain; it may yet show a larger effect. The cross-architecture portion of H_3 also remains future work.

The apparent reversal reported in earlier versions of this work was a confound. Because the original two experiments varied scarcity and filtering together, the sign flip looked like a filtering effect. The controlled factorial (Section 6.4) shows it is almost entirely a scarcity effect. The honest correction is that the filtered conditions' benefit derives overwhelmingly from the greater headroom at 25 images/class rather than from the filter, which adds at most a small, borderline increment — close to the alternative the earlier draft flagged as plausible and could not yet rule out.

The practical lesson for low-resource domains — including the medical imaging setting this study is a proxy for — is two-sided, but not the lesson the earlier draft drew. Generative augmentation is not regime-neutral. From a converged, visually plausible generator, it measurably harms a classifier that already has enough data, and measurably helps one that does not — by a large margin. The decision that matters most is whether you are genuinely data-starved; Stage 1 curation of the synthetic pool adds only a small, borderline increment on top. The operational news is good: once you are in the scarce regime, plain unfiltered augmentation captures nearly all of the benefit. The simplest pipeline is therefore very nearly the best-supported one — at least until Stage 2 perceptual filtering is tested. *Diagnose the regime; if you are starved, augment — and keep it simple.*

Future work, in priority order: (1) Stage 2 of the filter (SSIM/LPIPS perceptual screening) — the factorial tests Stage 1 critic ranking only; Stage 2 perceptual gating is the most direct next test of whether quality filtering can produce a downstream accuracy gain; (2) a retention-threshold and pool-size sweep to map the quality–quantity tradeoff (the factorial fixes a single aggressive top-k operating point over a single 800/class pool); (3) more seeds and multiple splits to tighten power on the sub-point filter effect the present design cannot exclude; (4) the original C-VAE / C-GAN / C-VAEGAN comparison under the identical protocol, testing H_1; (5) the FID/SSIM/LPIPS metric suite computed for high- versus low-scoring samples; (6) transfer to a genuine medical imaging task and to a curated CASIA-HWDB subset. The 2×2 factorial that earlier drafts listed as the top priority is the controlled experiment now reported in Section 6.4.

8. Computational Resources Used

All experiments ran on a single NVIDIA L4 Graphics Processing Unit (GPU) via Google Colab Pro. The dominant cost was the 500-epoch C-WGAN-GP training run (≈3.5 hours). The 100-epoch pilot, thirty CNN classifier training runs (six conditions × five seeds, 2–4 minutes each), and synthetic generation brought the total to approximately 8–10 GPU-hours. This is well under

the 120 GPU-hours estimated at midpoint for the full three-architecture study, consistent with the descoped design. Model checkpoints were saved to persistent storage every 25 epochs, enabling recovery across session interruptions and exact reuse of the trained generator and critic across both experiments. All code is PyTorch; the complete pipeline, with per-cell documentation and all executed outputs, accompanies this paper as a Jupyter notebook.

Acknowledgements

This paper emerged from several months of sustained, iterative dialogue with Claude (Anthropic), used here as a research thinking partner rather than a writing tool. The core research question, the choice of Chinese characters as a domain, the three-model comparative framing, and the experimental hypotheses all originated in my own thinking. Claude's role was to sharpen those ideas through rigorous back-and-forth, flag gaps in reasoning, suggest implementation strategies, and help translate rough intuitions into precise technical language. The architecture decisions, the quality filter pipeline, and the dataset selection rationale each went through multiple rounds of challenge and refinement in that dialogue. I find this mode of working — bringing your own ideas and using AI to stress-test and develop them — to be a genuinely productive form of intellectual partnership, and wanted to acknowledge it honestly rather than obscure it.

References

[1] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," in Proc. ICLR, 2014. arXiv:1312.6114.

[2] K. Sohn, H. Lee, and X. Yan, "Learning Structured Output Representation using Deep Conditional Generative Models," in Proc. NeurIPS, vol. 28, 2015.

[3] I. J. Goodfellow et al., "Generative Adversarial Nets," in Proc. NeurIPS, vol. 27, 2014. arXiv:1406.2661.

[4] M. Mirza and S. Osindero, "Conditional Generative Adversarial Nets," arXiv:1411.1784, 2014.

[5] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther, "Autoencoding beyond Pixels using a Learned Similarity Metric," in Proc. ICML, 2016. arXiv:1512.09300.

[6] B. Kong and Y. Xu, "Generative Adversarial Networks for Chinese Character Image Synthesis," unpublished manuscript, 2021.

[7] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved Training of Wasserstein GANs," in Proc. NeurIPS, vol. 30, 2017. arXiv:1704.00028.

[8] C. Zhang, P. Yin et al., "Chinese Character Recognition with Deep CNNs," in Proc. ICDAR, 2017.

[9] C.-L. Liu, F. Yin, D.-H. Wang, and Q.-F. Wang, "CASIA Online and Offline Chinese Handwriting Databases," in Proc. ICDAR, 2011.

[10] Chinese-MNIST Dataset. Kaggle. [Online]. Available: https://www.kaggle.com/datasets/gpreda/chinese-mnist.

[11] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image Quality Assessment: From Error Visibility to Structural Similarity," IEEE Trans. Image Process., vol. 13, no. 4, pp. 600–612, 2004.

[12] R. Zhang et al., "The Unreasonable Effectiveness of Deep Features as a Perceptual Metric," in Proc. CVPR, 2018. arXiv:1801.03924.

[13] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," in Proc. ICML, 2017. arXiv:1701.07875.