7. M. Abadi, P. Barham, J. Chen, et al. TensorFlow: A System for Large-Scale Machine Learning. In Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI), pages 265–283, 2016.

8. Keras API documentation [Electronic resource]. URL: https://keras.io/.

9. F. Pedregosa, G. Varoquaux, A. Gramfort, et al. Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, vol. 12, pages 2825–2830, 2011.

10. Streamlit documentation [Electronic resource]. URL: https://docs.streamlit.io/.

11. N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. Journal of Machine Learning Research, vol. 15, pages 1929–1958, 2014.

12. T. Dozat. Incorporating Nesterov Momentum into Adam. In International Conference on Learning Representations, Workshop Track, 2016.

УДК 004.932

QUATERNION CONVOLUTIONS IN LARGE-MASK IMAGE INPAINTING

Дмитро Колодочка, Марина Полякова

Національний університет «Одеська політехніка», м. Одеса, Україна

Introduction. Image inpainting is the process of restoring missing or corrupted image regions so that the result is visually plausible and semantically consistent with its surroundings. The problem splits into two broad regimes. In large-mask inpainting the unknown regions are wide and reconstruction depends on long-range propagation of structure and color. In the fine-grained missing-pixel regime many pixels are absent but the gaps between known pixels stay small. Large-mask inpainting is the harder and more practically relevant case: it is used in tasks like object removal, film rig removal, and region editing in generative systems, where large gaps require semantic reconstruction of background, geometry, and lighting rather than simple texture copying.

Recent quaternion inpainting work covers different missing-pattern regimes: Miao et al. [1] study a matrix-completion setting with evenly distributed missing pixels and small local gaps, while Wang et al. [2] explore hole-like masks closer to large-mask inpainting. Building on that direction, we study quaternion convolution (QConv) inside a LaMa-style architecture rather than a standard QConv GAN, and push the evaluation toward larger mask gaps. We chose LaMa-Fourier because its Fast Fourier Convolution (FFC) design [3] targets wide missing regions and global context, making it a good baseline for studying QConv beyond local completion.

QConv treats RGB information as a coupled entity rather than three independently mixed scalar channels. This may improve color consistency in broad smooth regions, but stronger channel coupling may reduce the flexibility needed for fine local texture. We therefore expect to measure a trade-off between color consistency and texture detail.

Aim. The aim of this work is to introduce selective QConv components into a LaMa-Fourier generator and to measure their effect on reconstruction quality, inference speed, and model size. Training stays fixed. Only the locations at which QConv replaces standard convolutions in the up-scaling, down-scaling, and local FFC paths are varied. We report FID, LPIPS, PSNR, SSIM, parameter count, throughput, and qualitative behavior on thin, medium, wide, and ultrawide masks, against a LaMa-Fourier baseline trained under the same data and optimization settings.

Materials and methods. The baseline generator follows LaMa's FFC design, which is strong for large-mask completion because global context is injected early through Fourier operators [3]. The LaMa-Fourier generator (Fig. 1) takes a three-channel masked image and a binary mask, uses a 7x7 convolution that lifts the input to 64 channels, downsamples through three 2D convolutions, processes the result with a stack of residual FFC blocks, upsamples through three transpose convolutions, and shrinks the image back to 3 channels with a 7x7 transpose convolution. Each FFC block splits features into local and global branches

and applies Fourier transform in the global branch. The path-specific modules LocalConvLayer, UpsampleConvLayer, DownsampleConvLayer are the blocks we vary across variants.

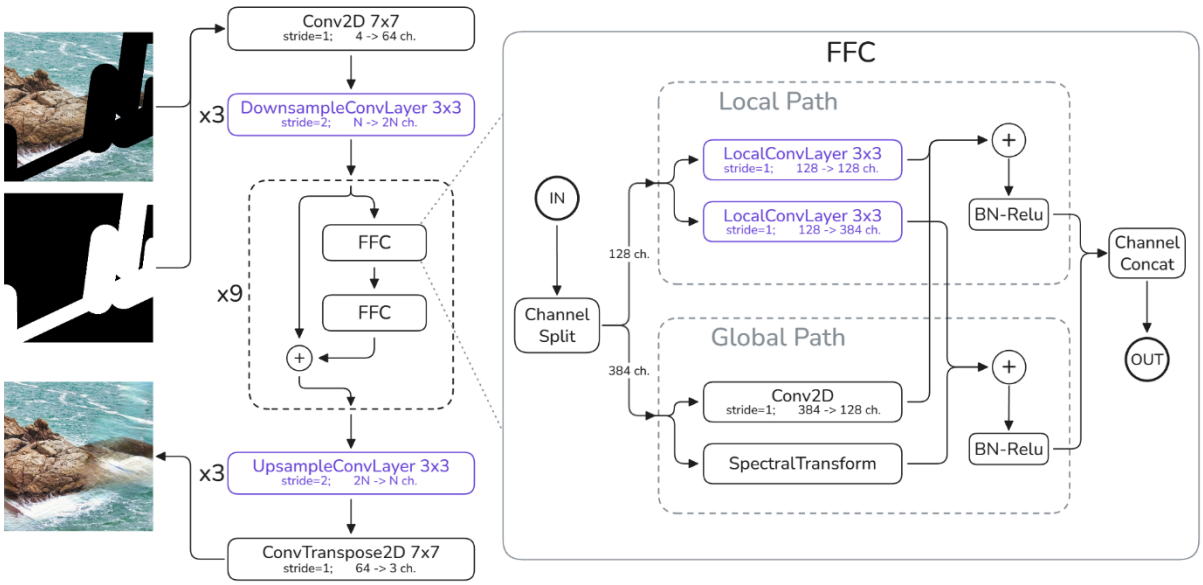


Fig. 1. Generator architecture. The Downsample, Upsample, and Local convolution layers vary across model variants.

Quaternion convolution. A quaternion is a hypercomplex number $q = q_0 + q_1 i + q_2 j + q_3 k$ with one real part and three imaginary parts, where i, j, k are the imaginary units of the quaternion algebra. The product of two quaternions is the Hamilton product, which mixes all four components. For $y = qx$ the four output components are:

$$y_0 = q_0 x_0 - q_1 x_1 - q_2 x_2 - q_3 x_3$$
$$y_1 = q_0 x_1 + q_1 x_0 + q_2 x_3 - q_3 x_2$$
$$y_2 = q_0 x_2 + q_2 x_0 + q_3 x_1 - q_1 x_3$$
$$y_3 = q_0 x_3 + q_3 x_0 + q_1 x_2 - q_2 x_1$$

A quaternion convolution [4] applies this coupling between four real-valued kernels and four input channel groups, so every kernel component interacts with every input component. This captures structured cross-channel dependencies that standard real-valued convolution does not model directly. It is also parameter-efficient: a quaternion layer learns only four kernels and defines the rest algebraically, reducing the parameter count.

Our quaternion convolutions implementation groups feature channels into tuples of four real channels, following the standard decomposition used in deep quaternion networks [5], so any tensor processed by a QConv layer must have a channel count divisible by four. Each layer is parameterized by four real-valued kernels coupled through the Hamilton product.

Training setup. We keep the training setup fixed and modify only the locations at which QConv replaces standard convolutions. LaMa-Fourier is the baseline with standard Conv2D/ConvTranspose2D modules, and each QConv variant changes a specific routing choice: Q-Up (QConv in UpsampleConvLayer only); Q-Down (DownsampleConvLayer only); Q-Local (LocalConvLayer only); Q-UpDown (UpsampleConvLayer and DownsampleConvLayer); and Q-DownLocal (DownsampleConvLayer and LocalConvLayer).

Training used the 1.44M-image training split of Places365 with runtime validation on the remaining split. All models were trained by us on random 256x256 crops resampled each epoch, with a batch size of 30 for 16 epochs, using the same discriminator, loss computation and hyperparameters as the original LaMa study, with learning rates of $lr_G = 10^{-3}$ and $lr_D = 10^{-4}$ for generator and discriminator respectively. The

LaMa-Fourier baseline was trained and evaluated on this dataset, rather than taken from a published checkpoint.

Evaluation setup. For final evaluation we used a separate fixed set of 1000 images randomly sampled from the LHQ-1024 dataset. For each mask preset, every image is processed with 9 pre-generated masks. FID is reported over those image-mask pairs and PSNR, SSIM, and LPIPS are aggregated over them, computed on the full output against the full ground truth using each run's last checkpoint. Masks were generated in four presets that differ mainly in target coverage and stroke width: thin (narrow irregular masks, 20-30% area), medium (irregular and box masks, 35-45%), wide (larger strokes, 50-60%), and ultrawide (70-80%). Examples are shown in Fig. 2.

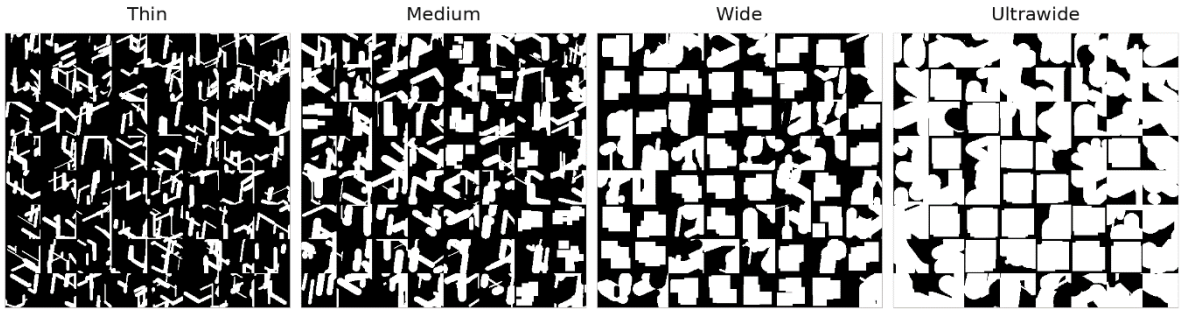


Fig. 2. Example masks for the four evaluation presets. 64 masks per image.

Results and discussion. Tables 1 and 2 report the metrics for each variant on 256x256 images, broken down by mask preset. Throughput is the mean batch-size-1 forward speed measured over 256 timed passes on an Intel Core i7-13700K CPU and a single NVIDIA RTX 4080 SUPER GPU workstation. Compared to LaMa-Fourier, every QConv model uses fewer parameters, is slower, and has lower PSNR. FID worsens on thin and medium masks and improves on ultrawide masks; on wide masks only the local-path models improve. SSIM rises for the scale-path placements and falls when QConv is used in the local path.

Table 1. FID, LPIPS, parameters, and throughput per variant and mask preset.

Model	FID Thin	FID Med	FID Wide	FID Ultra	LPIPS Thin	LPIPS Med	LPIPS Wide	LPIPS Ultra	Params M	Speed img/s
LaMa-Fourier	**8.21**	**14.98**	37.50	84.32	**0.087**	0.161	0.266	0.417	27.05	**37.58**
Q-Up	10.43	18.66	38.54	74.20	0.091	0.165	0.272	0.422	25.89	35.49
Q-Down	9.12	16.08	38.56	81.19	0.092	0.162	**0.261**	**0.400**	25.89	36.58
Q-Local	8.47	15.43	**34.99**	73.96	0.090	0.164	0.270	0.426	19.08	27.47
Q-UpDown	10.56	18.64	37.82	78.60	0.095	0.169	0.276	0.433	24.73	33.08
Q-DownLocal	8.49	16.17	36.49	**69.99**	**0.087**	**0.160**	0.273	0.443	**17.92**	24.67

Table 2. PSNR and SSIM per variant and mask preset.

Model	PSNR Thin	PSNR Med	PSNR Wide	PSNR Ultra	SSIM Thin	SSIM Med	SSIM Wide	SSIM Ultra
LaMa-Fourier	**27.42**	**24.07**	**20.91**	**17.76**	0.877	0.796	0.691	0.553
Q-Up	27.09	23.86	20.68	17.58	**0.885**	**0.805**	**0.703**	**0.567**
Q-Down	27.04	23.90	20.83	17.74	0.881	0.800	0.697	0.559
Q-Local	27.13	23.94	20.84	17.68	0.870	0.789	0.684	0.534
Q-UpDown	26.72	23.63	20.59	17.39	0.876	0.795	0.693	0.551
Q-DownLocal	27.33	23.99	20.54	17.01	0.884	0.802	0.695	0.541

Q-Up worsens FID and LPIPS but raises SSIM the most. Q-Local reaches the best wide-mask FID and lowers SSIM on every preset. Q-Down improves SSIM and is the only model that improves LPIPS on wide and ultrawide masks. Q-DownLocal shows the lowest ultrawide-mask FID and the smallest parameter count, with the largest speed and PSNR cost.

The qualitative samples show a trade-off in high and low frequency image detail (Fig. 3). Up-scaling QConv often introduces grid-like structures and blurrier reconstructions near image boundaries, and local-path QConv degrades texture further. Up-scaling QConv can still produce more coherent color in smooth regions such as sky. Q-Down shows the strongest behavior: it reconstructs straighter large structures interrupted by wide masks and preserves color continuity over large missing areas. Q-Up sometimes introduces a weak color tint into muted regions.

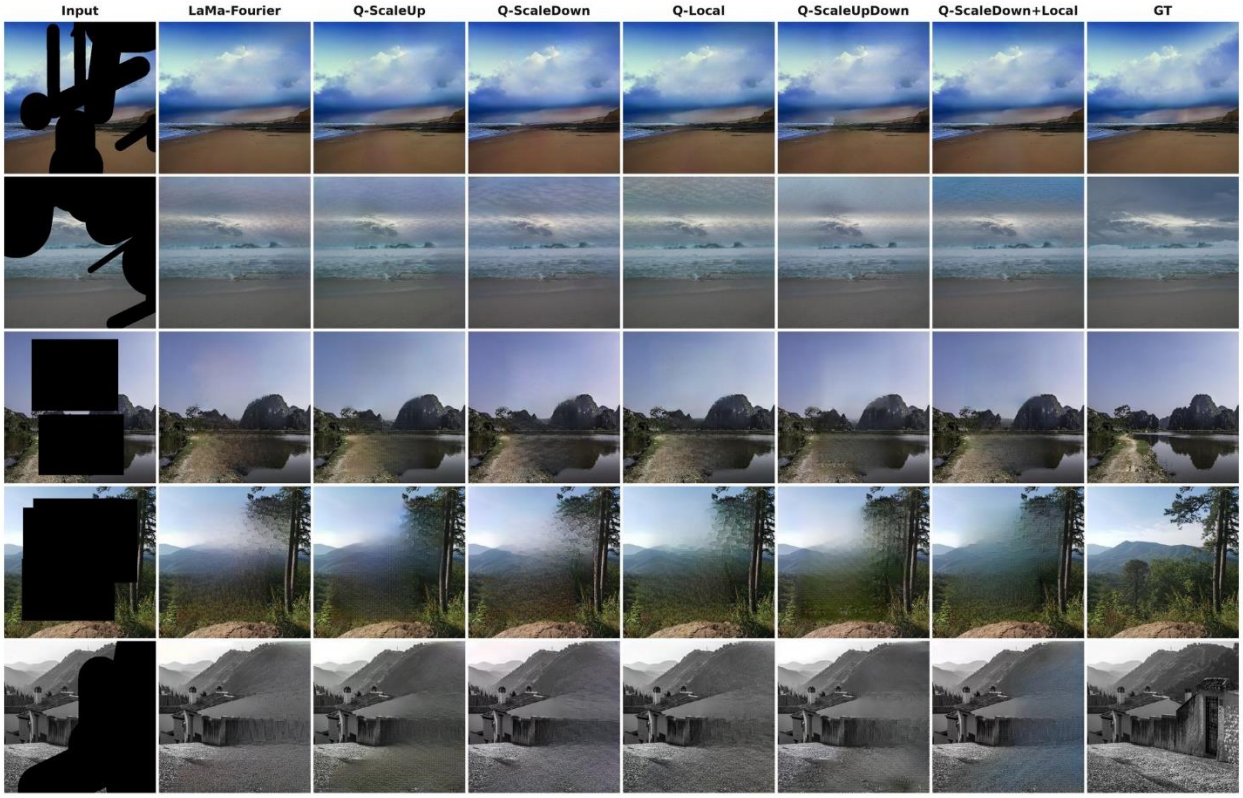


Fig. 3. Wide-mask result samples for baseline and studied model variants.

Taken together, the outcome depends on placement. Q-Down is the strongest large-mask placement, while Q-Up gives the highest SSIM and is weaker on FID and LPIPS. Q-DownLocal cuts the most parameters and is the slowest. Fewer parameters make QConv attractive for memory-constrained, non-real-time deployments. The main limitations are a single training regime inherited from LaMa and training on Places365 only. Tuning the loss per variant and validating on additional datasets are promising directions for future work.

Conclusions. Selective QConv integration in LaMa-style inpainting is feasible, and the effect depends on placement. Up-scaling QConv raises SSIM, down-scaling QConv is the most balanced large-mask placement, and local-path QConv improves wider-mask FID while lowering SSIM. The reduced parameter count points to a lower memory footprint, at the cost of slower inference and lower PSNR. The most promising direction is to keep these placement-specific gains while narrowing the LPIPS and PSNR gap.

References

1. Miao J., Kou K. I., Yang L., Han J. Quaternion matrix completion using untrained quaternion convolutional neural network for color image inpainting. arXiv.org. 2023. URL: https://arxiv.org/abs/2305.00416 (дата звернення: 16.06.2026).

2. Wang D., Zhu D., Zhao M., Jia Z. Quaternion generative adversarial neural networks and applications to color image inpainting. arXiv.org. 2024. URL: https://arxiv.org/abs/2406.11567 (дата звернення: 16.06.2026).

3. Suvorov R., Logacheva E., Mashikhin A. et al. Resolution-robust large mask inpainting with Fourier convolutions. Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). Waikoloa, 2022. P. 2149–2159.

4. Zhu X., Xu Y., Xu H., Chen C. Quaternion convolutional neural networks. Proceedings of the European Conference on Computer Vision (ECCV). Munich, 2018. P. 631–647.

5. Gaudet C. J., Maida A. S. Deep quaternion networks. 2018 International Joint Conference on Neural Networks (IJCNN). Rio de Janeiro, 2018. P. 1–8.

УДК 338.48:004.8

ПСИХОЛОГІЧНІ АСПЕКТИ ВИКОРИСТАННЯ ШТУЧНОГО ІНТЕЛЕКТУ У СФЕРІ ГОСТИННОСТІ

Дар'я Гречко

Миколаївський національний аграрний університет, м. Миколаїв, Україна

Сучасна індустрія гостинності активно трансформується під впливом цифрових технологій. Готелі, ресторани та інші підприємства сфери гостинності використовують електронні системи бронювання, мобільні сервіси, автоматизовані системи управління, CRM-технології, чат-боти та засоби аналізу даних. Особливого значення набуває штучний інтелект (ШІ), можливості якого дозволяють автоматизувати окремі операції, аналізувати значні масиви інформації та вдосконалювати взаємодію з гостями.

Використання ШІ у сфері гостинності має не лише технологічний та економічний, а й психологічний вимір. Результативність упровадження інтелектуальних технологій значною мірою залежить від того, як їх сприймають гості та працівники. Особливого значення набувають довіра до технології, готовність взаємодіяти з автоматизованими системами, ставлення до заміщення традиційного обслуговування цифровим, а також психологічна адаптація персоналу до технологічних змін.

Одним із ключових психологічних чинників прийняття ШІ споживачами є довіра. Дослідження O. H. Chi, C. G. Chi, D. Gursoy та R. Nunkoo показує, що довіра до взаємодії з роботами на основі ШІ істотно впливає на намір клієнтів користуватися такими технологіями у сфері гостинності [1]. Отже, впровадження ШІ потребує не лише технічної надійності системи, але й формування позитивного користувацького досвіду та зрозумілої взаємодії між клієнтом і технологією. Водночас технологічна зручність не усуває потреби в людському компоненті обслуговування. Для сфери гостинності важливо поєднувати можливості автоматизації з професійною комунікацією персоналу, особливо коли ситуація потребує індивідуального підходу, пояснення або емоційної підтримки гостя. Такий підхід дає змогу використовувати переваги цифрових технологій без повної заміни безпосередньої взаємодії між працівником і клієнтом.

Використання ШІ в закладах індустрії гостинності також пов'язане з автоматизацією бізнес-процесів. Н. А. Антошкова, Л. О. Гризовська та Н. В. Прилепа розглядають ШІ як технологію автоматизації закладів готельно-ресторанного бізнесу та підкреслюють його роль в інтеграції інформаційних систем і технологій підприємств гостинності [2]. Автоматизовані системи доцільно використовувати для стандартизованих і повторюваних операцій, тоді як у нестандартних і конфліктних ситуаціях особливого значення набувають індивідуальний підхід, гнучкість комунікації та людська взаємодія.

Окремим напрямом використання ШІ є персоналізація обслуговування. Ю. О. Шейко визначає штучний інтелект як перспективний інструмент персоналізації послуг індустрії гостинності та розглядає його використання для кращого розуміння і задоволення потреб клієнтів на різних етапах обслуговування [3]. Цифрові системи можуть опрацьовувати інформацію про попередню взаємодію