The Mechanics of CNN Filtering with Rectification

Supplementary Material

Here we provide additional results and visualizations to accompany our work in characterizing the mechanics the rectified convolution operation, demonstrate how even and odd filter components act upon image information as rest and kinetic energy operators, respectively, where the velocity of directional information is determined by the ratio of kinetic to total filter energy, i.e. the ratio of odd to total energy.

In Section A we provide definitions of even and odd symmetry for 2D filters.

In Section B we demonstrate the role of DCT coefficients and the dominance of primary DC and gradient components $\Sigma, \nabla_x, \nabla_y$ in training from scratch accuracy in spectral energy distributions, from VGG and Resnet models.

In Section C we provide additional demonstrations of our information propagation theory while mixing between even (e.g. DC Σ) and odd (e.g. gradient ∇_x, ∇_y) components, for various combinations of test patterns (pixel, circle), filter sizes (2x2, 3x3), types (DC, gradient, translation) and activation functions (none, ReLU, Modulus).

A. Even-Odd Symmetry for 2D Images

Our work assumes even functions defined by rotational symmetry on a 2D lattice, as characterized by the dihedral group, here we introduce definitions used. Let $f(x, y)$ be a discrete 2D image or kernel of size $N \times N$ pixels, defined as a mapping $f : \mathbb{Z}^2 \to \mathbb{R}^1$ from 2D coordinates $(x, y) \in \mathbb{Z}^2$ to a scalar value $f \in \mathbb{R}^1$. Any function $f(x, y)$ may be decomposed into a sum $f(x, y) = f_e(x, y) + f_o(x, y)$ of an even rotationally symmetric component $f_e(x, y)$ and an odd component $f_o(x, y)$ whose magnitudes follow a Pythagorean relationship as shown in Figure 10.

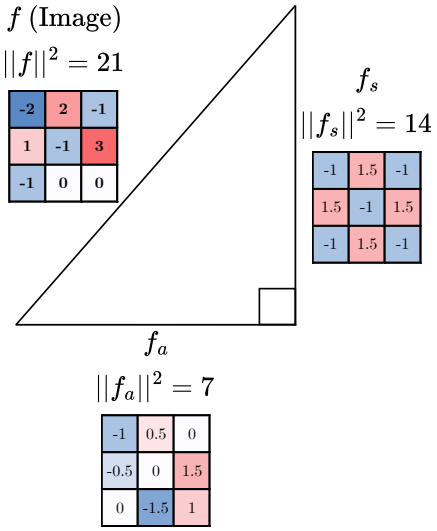


Figure 10. Illustrating the Pythagorean geometry of a discrete 2D image f (e.g. 3×3 kernel) decomposed into orthogonal odd f_o and even f_e components

Coordinates (x, y) are taken with respect to the image centre $(\frac{N-1}{2}, \frac{N-1}{2})$ without loss of generality and the primary properties of even and odd functions in 2D are as follows.

Definition 1 (Even (Symmetric) Image) *An image $f_e(x, y) \in \mathbb{R}^{n \times n}$ is rotationally symmetric if*

$$f_e(x, y) = f_e(\pm x, \pm y) = f_e(\pm y, \pm x),$$

where the unique value of $f_s(x, y)$ is the average of the set of equidistant points $\{(x, y) : r = \sqrt{x^2 + y^2}\}$ forming sign and coordinate permutations of (x, y):

$$f_s(x, y) = \frac{1}{8} \sum_{s_x, s_y \in \{\pm 1\}} f(s_x x, s_y y) + f(s_y y, s_x x) \tag{21}$$

Definition 2 (Odd (Anti-Symmetric) Image) *An image $f_o(x,y) \in \mathbb{R}^{n\times n}$ is rotationally anti-symmetric if*

$$f_o(x,y) = f(x,y) - f_e(x,y)$$

where the sum of $f_o(x,y)$ over the set of equidistant points $\{(x,y) : r = \sqrt{x^2+y^2}\}$ forming sign and coordinate permutations of (x,y) is 0:

$$\sum_{s_x,s_y\in\{\pm 1\}} f_o(s_x x, s_y y) + f_o(s_y y, s_x x) = 0 \tag{22}$$

Lemma 1 (Orthogonality) *Even and odd components are orthogonal and their scalar or dot product is thus 0:*

$$f_e(x,y)\cdot f_o(x,y) = \sum_{x,y} f_e(x,y) f_o(x,y) = 0$$

Definition 3 (Energy) *The energy of an image f is defined as the squared magnitude $\|f(x,y)\|^2$, which equals the sum of squared magnitudes of the even and odd components:*

$$\|f(x,y)\|^2 = \sum_{x,y} f^2(x,y) = \sum_{x,y} f_e^2(x,y) + \sum_{x,y} f_o^2(x,y).$$

B. Additional Training Results

Here we provide additional results and explanations regarding training experiments, even and odd components, and the discrete cosine transform (DCT) basis.

B.1. The DCT basis and Even and Odd filter components

Even and odd filters in 2D may generally take on a variety of unique patterns, *e.g.* DC Σ, gradients ∇ and higher order patterns for larger filter sizes. Here, we show how even and odd filters may be generally grouped as components of the discrete cosine transform (DCT) frequency transform, as is commonly done in image and video compression. In the following training experiments, we interpret the contribution of each DCT basis towards the task of classification.

In general, an $N \times N$-pixel filter may be represented as a sum of N^2 discrete cosine transform (DCT) coefficients, each of which may be purely even (Symmetric, S), purely odd (Antisymmetric, A) or mixed even + odd (M). Figure 11 a) shows the DCT basis functions up to index or wave number $(u,v) = (4,4)$. Figure 11 b) shows the pattern of symmetry ascribed each DCT basis. As shown in Figure 12, we may observe that where one or two wave numbers (u,v) are odd, the DCT basis is odd (A). If both indices (u,v) are even and equal $u = v$, *i.e.* along the main diagonal, the basis is even (S). If both indices (u,v) are even but unequal $u \neq v$, then the basis is mixed even and odd (M). Figure 12 shows the pattern of even (symmetric) and odd (antisymmetric) components for each DCT basis. To our knowledge, this is the first time the DCT has been expressed as even and odd components, despite the widespread use of the DCT in data compression.

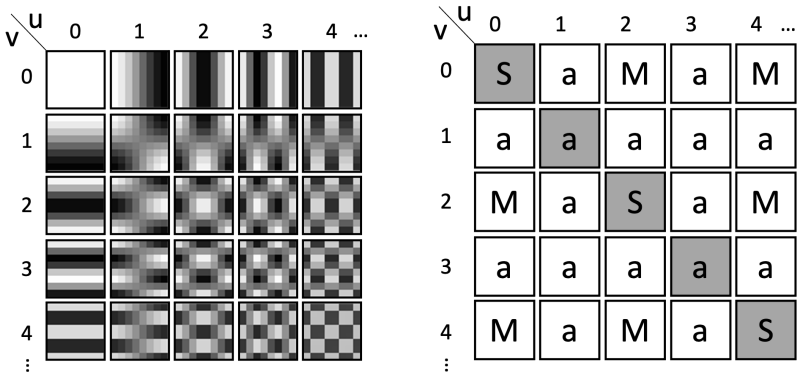
In practice, images or kernels representing natural images are dominated by low frequency DCT components. Particularly in our framework for small filters, *i.e.* 3×3 pixels, the even (or symmetric) component f_e may be approximated by DC or sum:

$$f_e(x,y) = \sum_{u,v\ \in\ Even} \omega_{u,v} D_{u,v} \approx \omega_{0,0} D_{0,0} = \omega_{0,0}\Sigma. \tag{23}$$

While the odd (or antisymmetric) component f_o may be approximated by the gradient or difference ∇, where a single angular parameter θ defines the gradient orientation as a linear mix between horizontal ∇_x and vertical ∇_y gradient components:

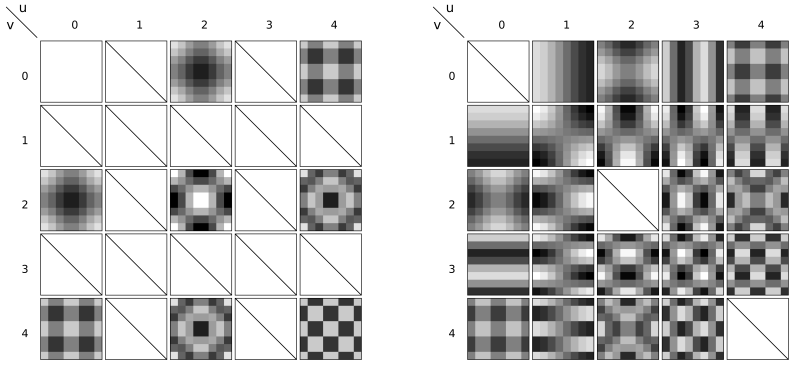
$$\begin{aligned} f_o(x,y) &= \sum_{u,v\ \in\ Odd\ \cup\ u\neq v} \omega_{u,v} D_{u,v}, \\ &\approx \omega_{0,1} D_{0,1} + \omega_{1,0} D_{1,0} \propto \cos\theta\nabla_x + \sin\theta\nabla_y. \end{aligned} \tag{24}$$

This approximation is validated in experiments, where retraining CNNs with only three of nine filter components $(\Sigma, \nabla_x, \nabla_y)$ results in greater than 90% of baseline accuracy for typical networks, e.g. VGG and Resnet.



a) DCT Bases b) Symmetry

Figure 11. a) shows the DCT bases up to index (4,4), b) indicates whether the basis is even (symmetric) S, odd (antisymmetric) a or mixed M



a) Even (symmetric) b) Odd (antisymmetric)

Figure 12. Illustrating the even a) and odd b) components of DCT bases, black and white indicate negative and positive values.

B.2. Training from Scratch on the CIFAR-100 Dataset using DCT components

This experiment consists of training a VGG16 [44] and Resent20 [24] models with various numbers of DCT coefficients, from a single DC (Σ) parameter to 9 total components (full spectrum). DCT kernel weight parameters $\{\omega_i\}$ are used, which are updated during backpropagation and inverse transformed into filters for forward propagation. For all 6 runs, we use the same hyperparameters which yielded the best accuracy for the baseline.

As we can see in Table 1, the run with only DC (Σ) and low-order gradient (∇_x, ∇_y) components, VGG16 achieves 91% of baseline validation accuracy. We observe the same behaviour when training Resnet20 (see Table 2).

Table 1. Training VGG16 on CIFAR-100 [29] using convolutional kernels composed of progressively additional high-order DCT components. We find that only 3 low-frequency components (underlined) contribute to 91% of VGG16 baseline accuracy.

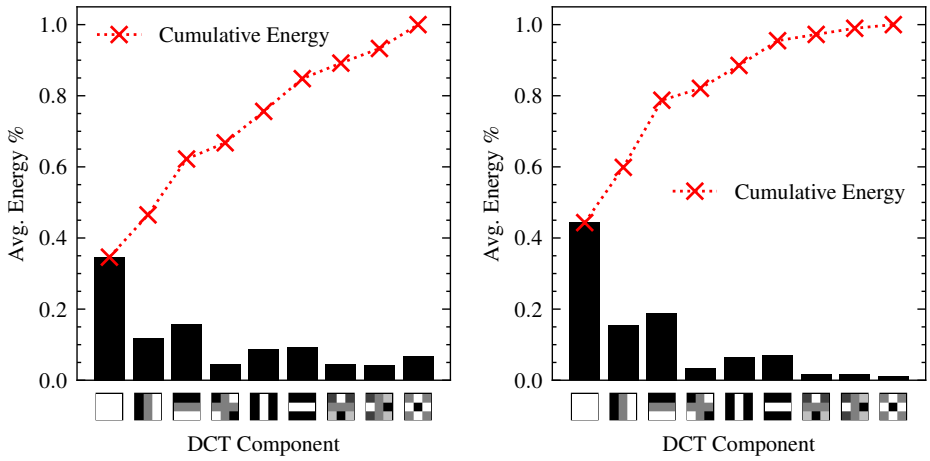
Number of DCT Components	Val-Accuracy ($\pm$ std)	% of Baseline
1 (Σ)	0.3247 $\pm$ 0.0052	0.44
<u>3 ($\Sigma, \nabla_x, \nabla_y$)</u>	<u>0.6664 $\pm$ 0.0017</u>	<u>0.91</u>
4	0.6823 $\pm$ 0.0039	0.93
6	0.7162 $\pm$ 0.0055	0.98
8	0.7294 $\pm$ 0.0032	0.99
9 (Baseline)	**0.7299 $\pm$ 0.0019**	**1.00**

Table 2. Training Resnet20 on CIFAR-100 [29] using convolutional kernels composed of progressively additional high-order DCT components. We find that only 3 low-frequency components (underlined) contribute to 92% of Resnet20 baseline accuracy.

Number of DCT Components	Val-Accuracy ($\pm$ std)	% of Baseline
1 (Σ)	0.4301 $\pm$ 0.022	0.63
<u>3 ($\Sigma, \nabla_x, \nabla_y$)</u>	<u>0.6277 $\pm$ 0.0025</u>	<u>0.92</u>
4	0.6413 $\pm$ 0.0084	0.94
6	0.6675 $\pm$ 0.0074	0.98
8	0.6759 $\pm$ 0.0069	0.99
9 (Baseline)	**0.6805 $\pm$ 0.0104**	**1.00**

B.3. The Energy of ImageNet-trained Kernels is concentrated into Σ and ∇ DCT components.

We report in Figure 14 and Figure 15 the average energy percentage for each frequency component ω_i of each kernel of VGG16 [44] and Resnet50 [24], respectively, trained on Imagenet [9] , across all layers. In Figure 13 we plot the average energy per spectral component ω_i for the entire network. As we can clearly see from Figure 13, the majority of the weights are either DC Σ (even) or gradient ∇ (odd) after training, whereas they are uniformly distributed across all components at initialization (Fig. 14).



(a) Resnet50 (b) VGG16

Figure 13. Spectral DCT decomposition ω_i of all 3×3 convolutional filters in all layers of (a)) Resnet50 and (b)) VGG16. We find that in both models the majority of the weights are comprised of low order DC and Gradients ($\Sigma + \nabla$)

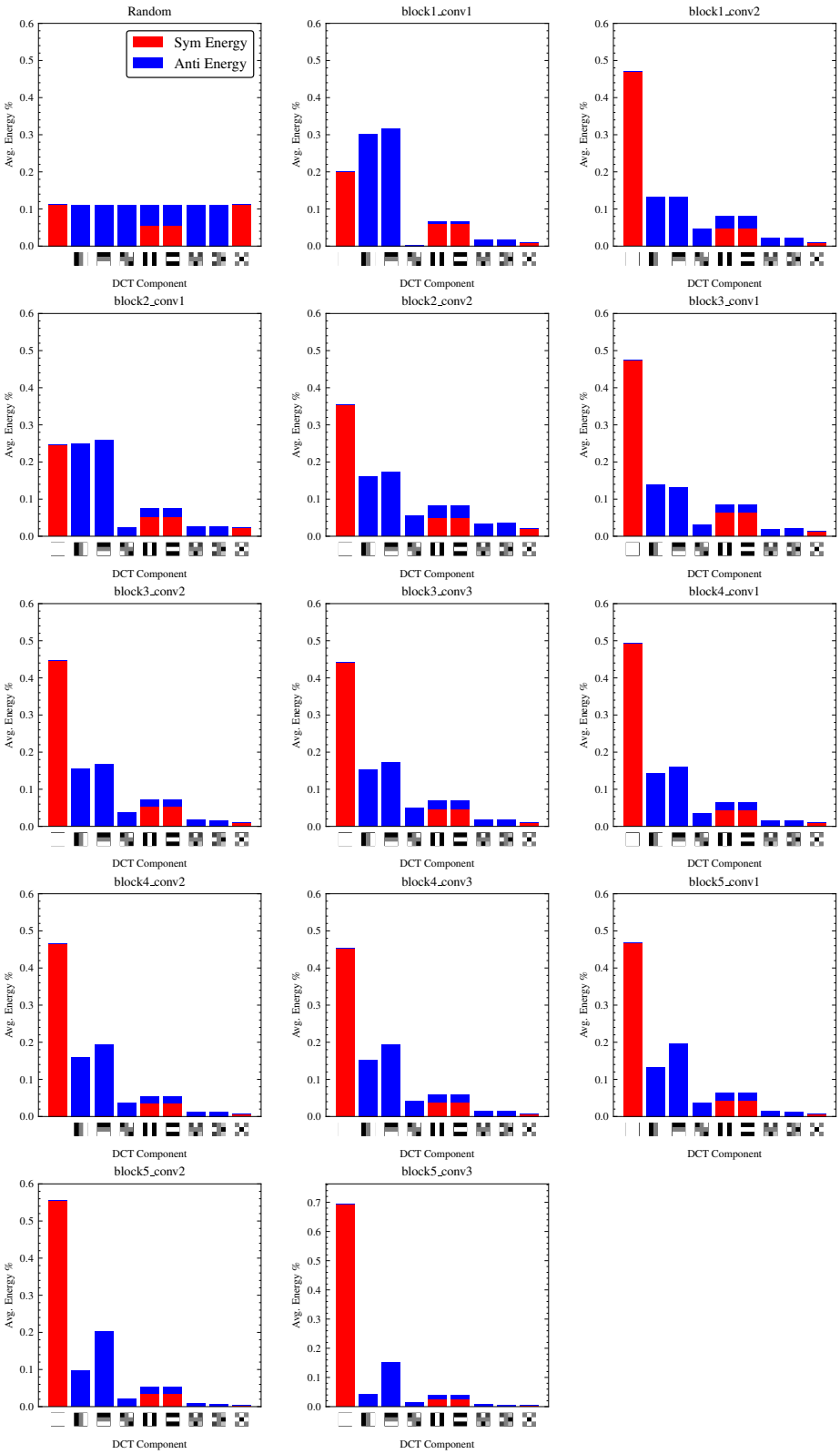


Figure 14. Average energy distribution of DCT components ($\frac{\omega_i^2}{||\boldsymbol{\omega}||^2}$) in random and learned convolutional kernels (trained on Imagenet) throughout VGG16 layers.

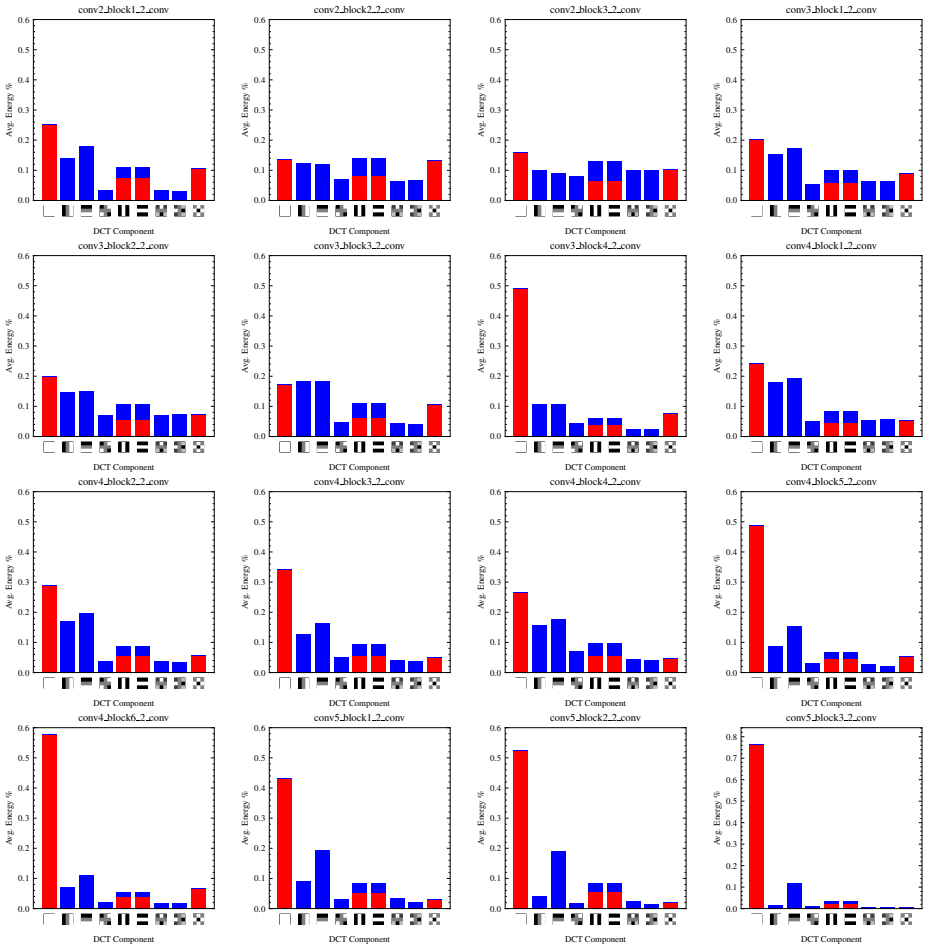


Figure 15. Average energy distribution of DCT components ($\frac{\omega_i^2}{||\omega||^2}$) in learned convolutional kernels (trained on Imagenet) throughout Resnet50 layers.

C. Additional Propagation Demonstrations

Here we demonstrate the results of rectified convolution from a single channel, and how the velocity of information is determined by the mixing ratio β of even and odd filter components, similarly to the Lorentz transform in the theory of special relativity. Rectified convolution is repeatedly performed upon test patterns, and the velocity is measured in terms of the displacement of the centre of mass per convolution.

C.1. Experimental Setup

Here we demonstrate the mechanics by which even (e.g. DC Σ) and odd (e.g. gradient ∇_x, ∇_y) filter components act upon image information, similarly to Section 4.3 of the paper, for various combinations of test patterns (pixel, circle), filter sizes (2×2, 3×3), types (DC, gradient, translation) and activation functions (none, ReLU, Modulus).

Table 3 shows the filter kernels used for various values of β^2. Most demonstrations mix Σ and ∇_x components according to the β^2 parameter, and convolve a test pattern. Note that 2x2 kernels are applied alternatingly within a 3x3 kernel in order to avoid a half-pixel shift following convolution. We also test a special case of propagation with a translation kernel, which is normally an offset impulse kernel (Table 3, 3×3 translation for $\beta^2 = 0.75$).

Our demonstrations perform rectified convolution on two test image patterns including a circle ($r = 19$) Fig. 16a) and an impulse (Fig. 16b). Between each iteration, the activation centre of mass μ_x and standard deviation σ_x are computed from a normalized activation $f(x, y)$ as follows:

$$\mu_x = \frac{\sum_x x \|f(x,0)\|}{\sum_x \|f(x,0)\|} \qquad \sigma^2 = \frac{\sum_x \|f(x,0)\|(x-\mu_x)^2}{\sum_x \|f(x,0)\|} \tag{25}$$

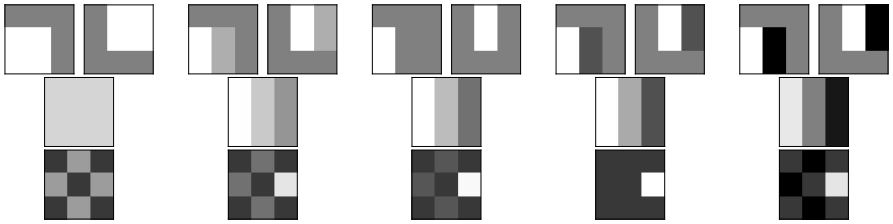
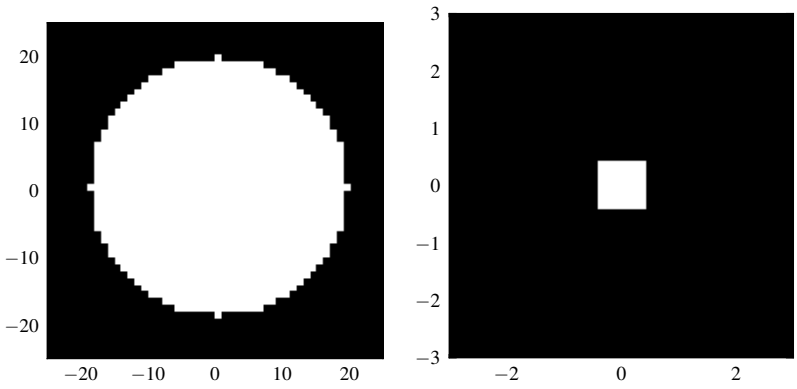
Kernel size	$\boldsymbol{\beta^2}$				
	0	0.25	0.5	0.75	1
2×2 (alternating)					
3×3					
3×3 (translation)					

Table 3. Examples of kernels used, where each 3×3 kernel $f = \beta \hat{f}_o + \sqrt{1-\beta^2}\hat{f}_e$ is generated by mixing odd f_o and even f_e components according to mixing ratio β.

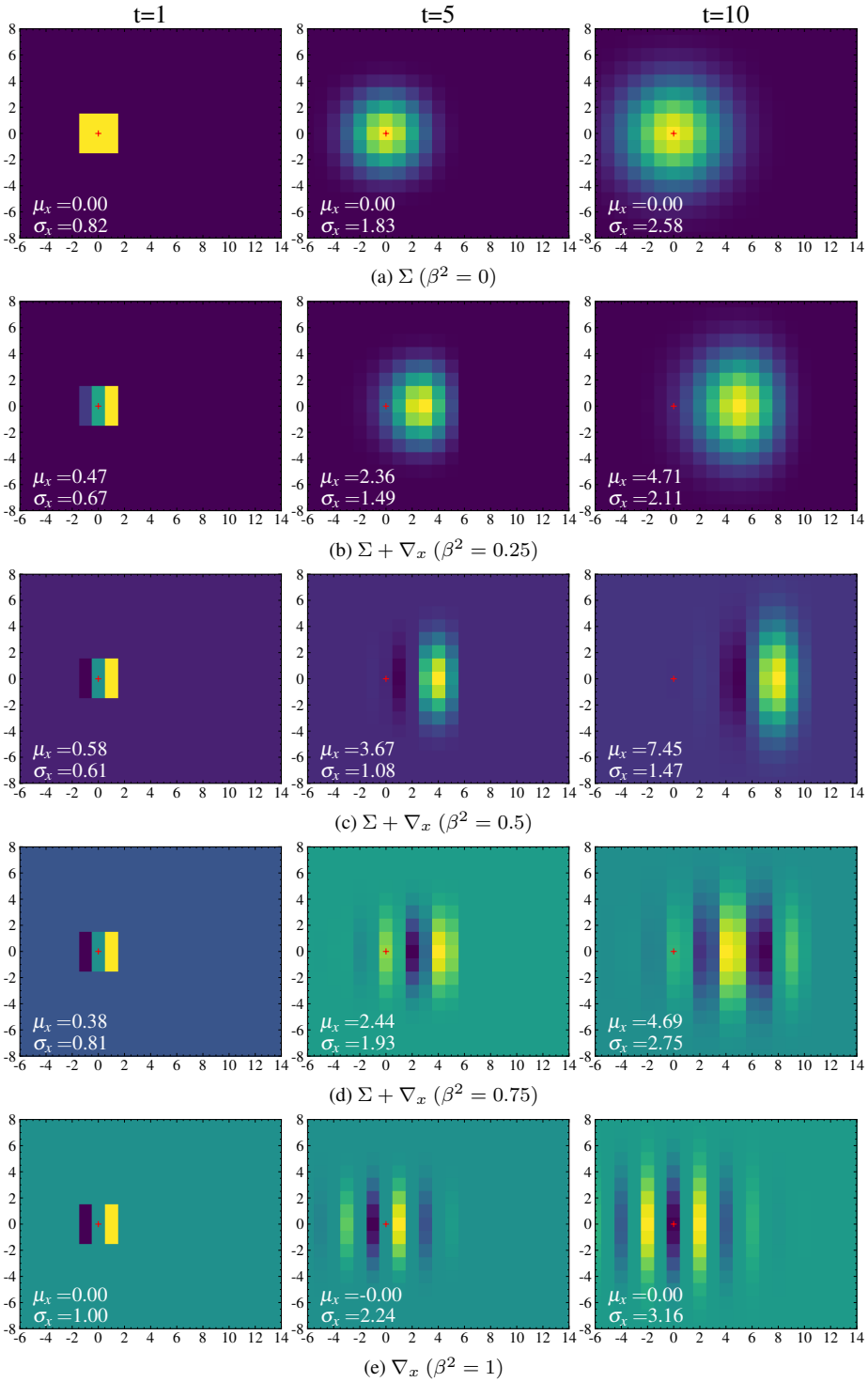


(a) Test Circle (r=19) (b) Test pixel (impulse)

Figure 16. Test Patterns

C.2. Convolution Without Activation

C.2.1. 3×3 kernel, mixing unidirectional gradient ∇_x and sum Σ



(a) Σ ($\beta^2 = 0$)

(b) $\Sigma + \nabla_x$ ($\beta^2 = 0.25$)

(c) $\Sigma + \nabla_x$ ($\beta^2 = 0.5$)

(d) $\Sigma + \nabla_x$ ($\beta^2 = 0.75$)

(e) ∇_x ($\beta^2 = 1$)

Figure 17. Demonstrating the effect of repeated convolution (no activation function) of a test pattern (pixel) over 3×3 kernels, varying β mixing between Σ and ∇. Note that for both $\beta = 0$ and $\beta = 1$, there is net displacement of the centre of mass, this is distinctly different from rectified convolution.

C.3. Convolution With ReLU (rectification) Activation

C.3.1. 3×3 kernel, mixing unidirectional gradient ∇_x and sum Σ components

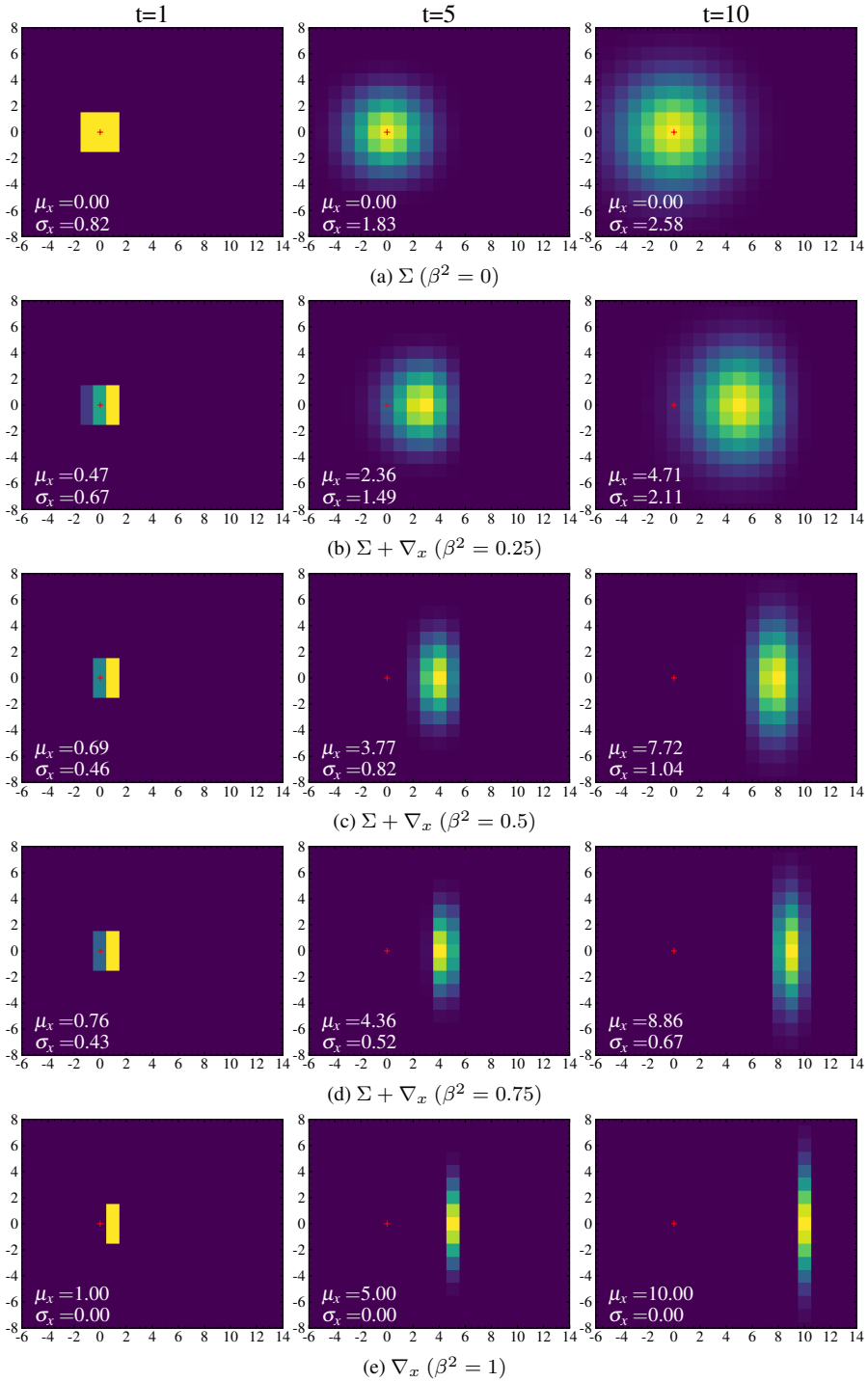
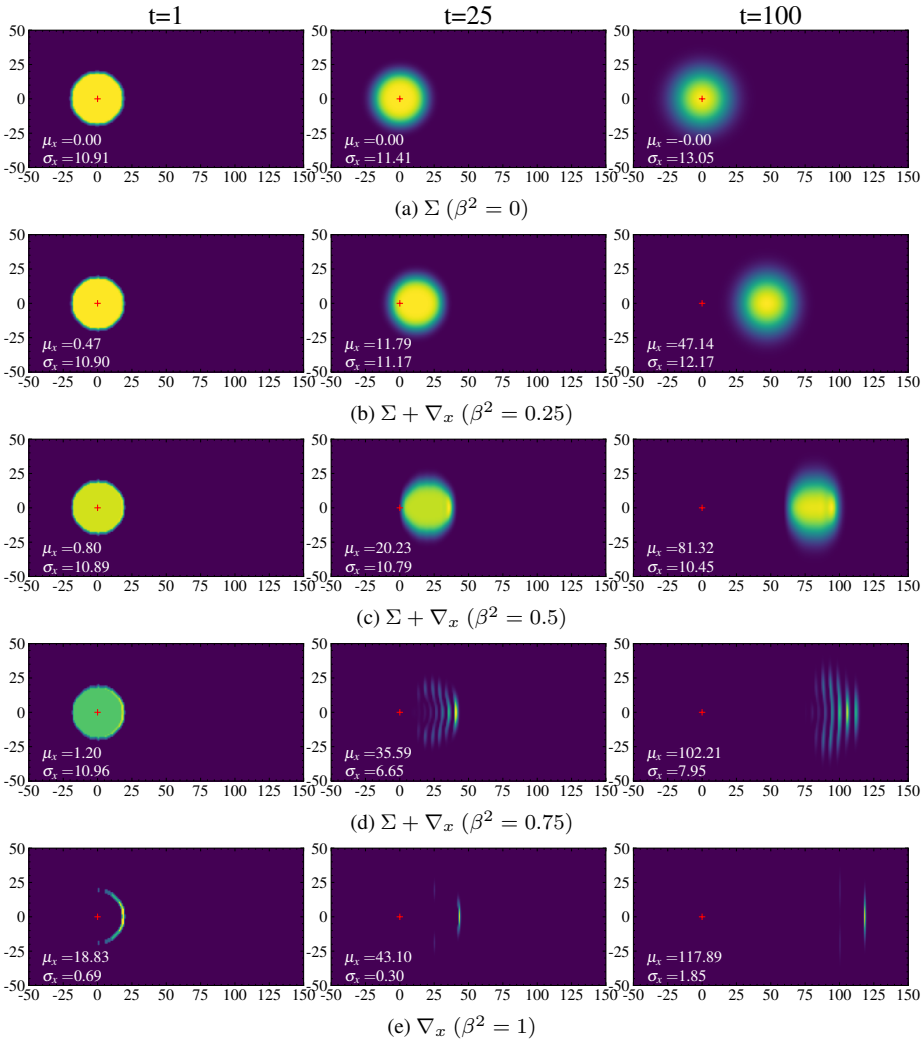


Figure 18. Demonstrating the effect of repeated convolution+ReLU of a test pattern over different types of 3×3 kernels (DC and Gradient). Note that for $\beta = 0$ a), content diffuses symmetrically about a stationary centre of mass, while for $\beta = 1$ the centre of mass translates rightward with maximum velocity.



(a) Σ ($\beta^2 = 0$)

(b) $\Sigma + \nabla_x$ ($\beta^2 = 0.25$)

(c) $\Sigma + \nabla_x$ ($\beta^2 = 0.5$)

(d) $\Sigma + \nabla_x$ ($\beta^2 = 0.75$)

(e) ∇_x ($\beta^2 = 1$)

Figure 19. Demonstrating the effect of repeated convolution+ReLU of a circular test pattern ($r = 19$) with 3×3 kernels mixing DC Σ and fixed direction gradient ∇_x over various mixing ratios $\beta \in \{0, 0.25, 0.5, 0.75, 1\}$. Note that for $\beta = 0$ a), content diffuses symmetrically about a stationary centre of mass, while for $\beta = 1$, the circle bulk disappears and the rightmost edge translates right with maximum velocity.

C.3.2. 3×3 kernel, translation filter components

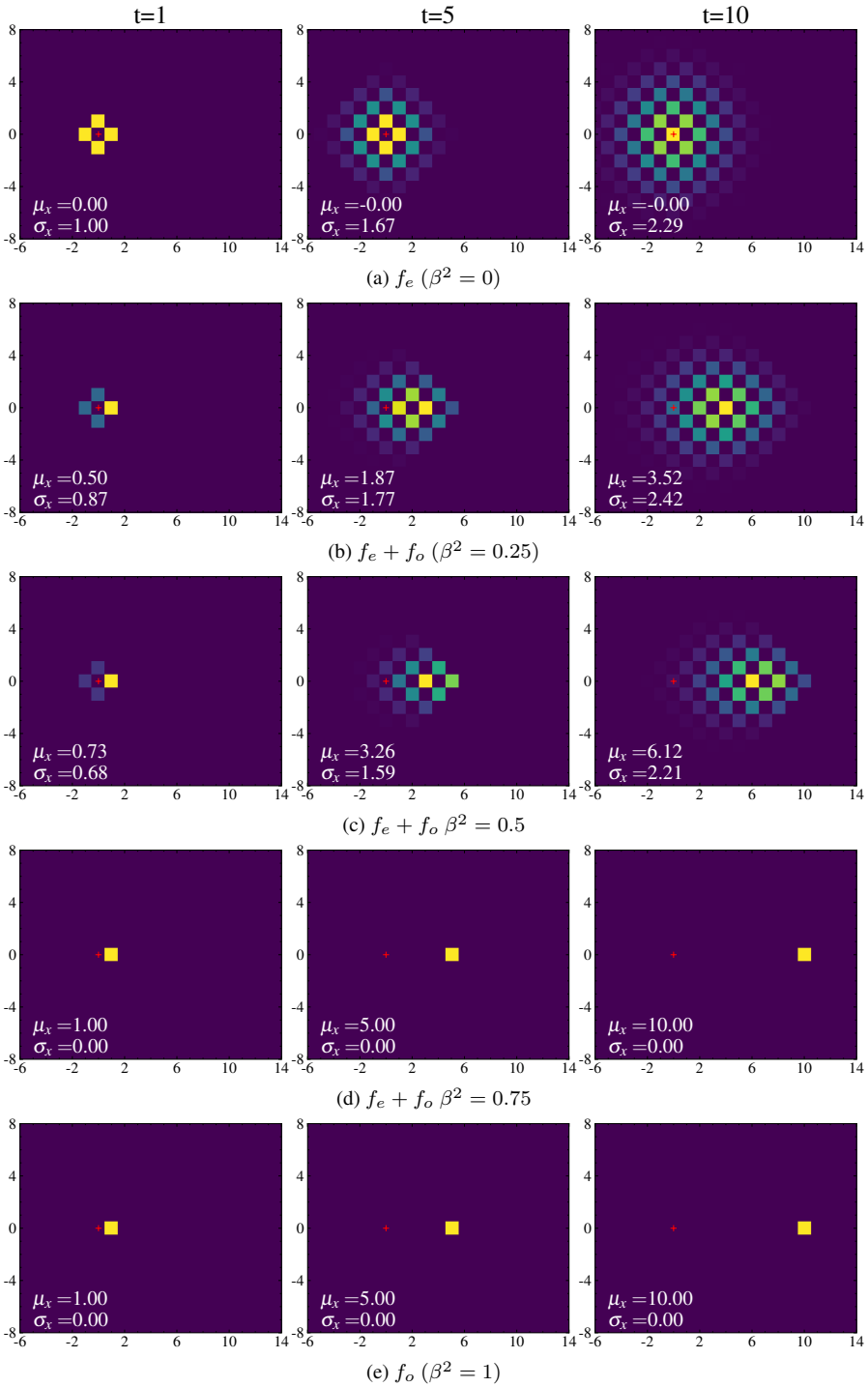
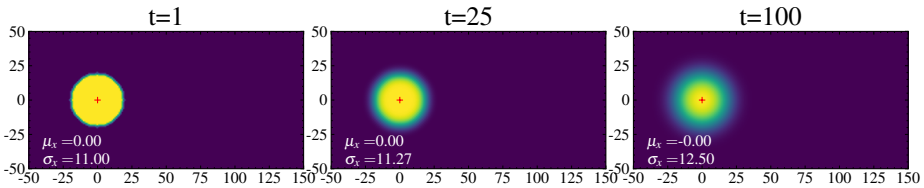
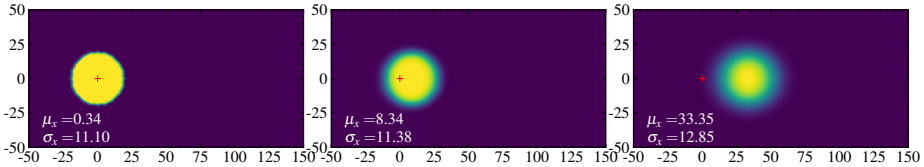


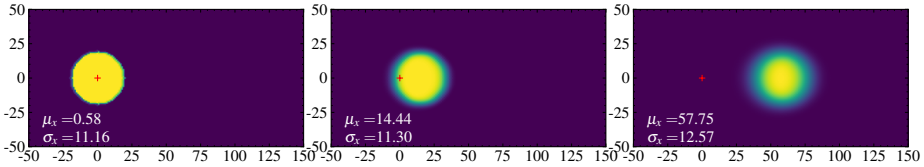
Figure 20. Demonstrating the effect of repeated convolution+ReLU of a test pattern (pixel) over 3×3 translations, varying β. Note that for $\beta = 0$ a), artificial checker-board structure appears due to the complex non-DC even (symmetric) component, while the content maintains a stationary centre of mass. For $\beta = 1$, the content translates rightward with maximum velocity.



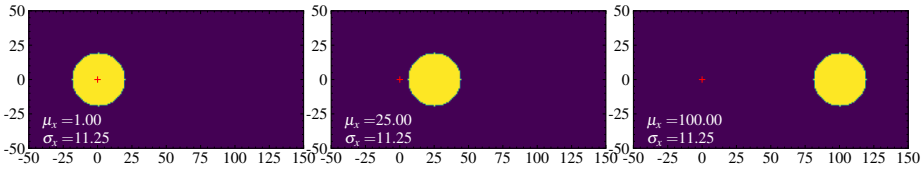
(a) $\Sigma\ \beta^2 = 0$



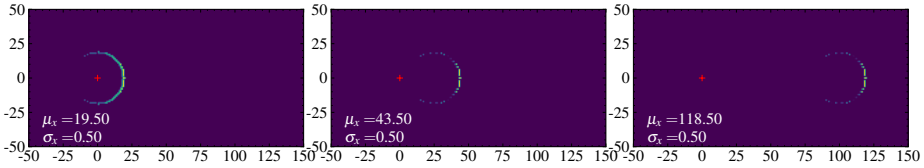
(b) $\Sigma + \nabla_x\ \beta^2 = 0.25$



(c) $\Sigma + \nabla_x\ \beta^2 = 0.5$



(d) $\Sigma + \nabla_x\ \beta^2 = 0.75$



(e) $\nabla_x\ \beta^2 = 1$

Figure 21. Demonstrating the effect of repeated convolution+ReLU of a test pattern (circle) over 3×3 translation filter components, varying β. Note that for $\beta = 1$, the circle bulk disappears and the rightmost edge translates rightward with maximum velocity.

C.3.3. 3×3 kernel, mixing alternating gradient $\{\nabla_x, -\nabla_x\}$ and sum Σ components

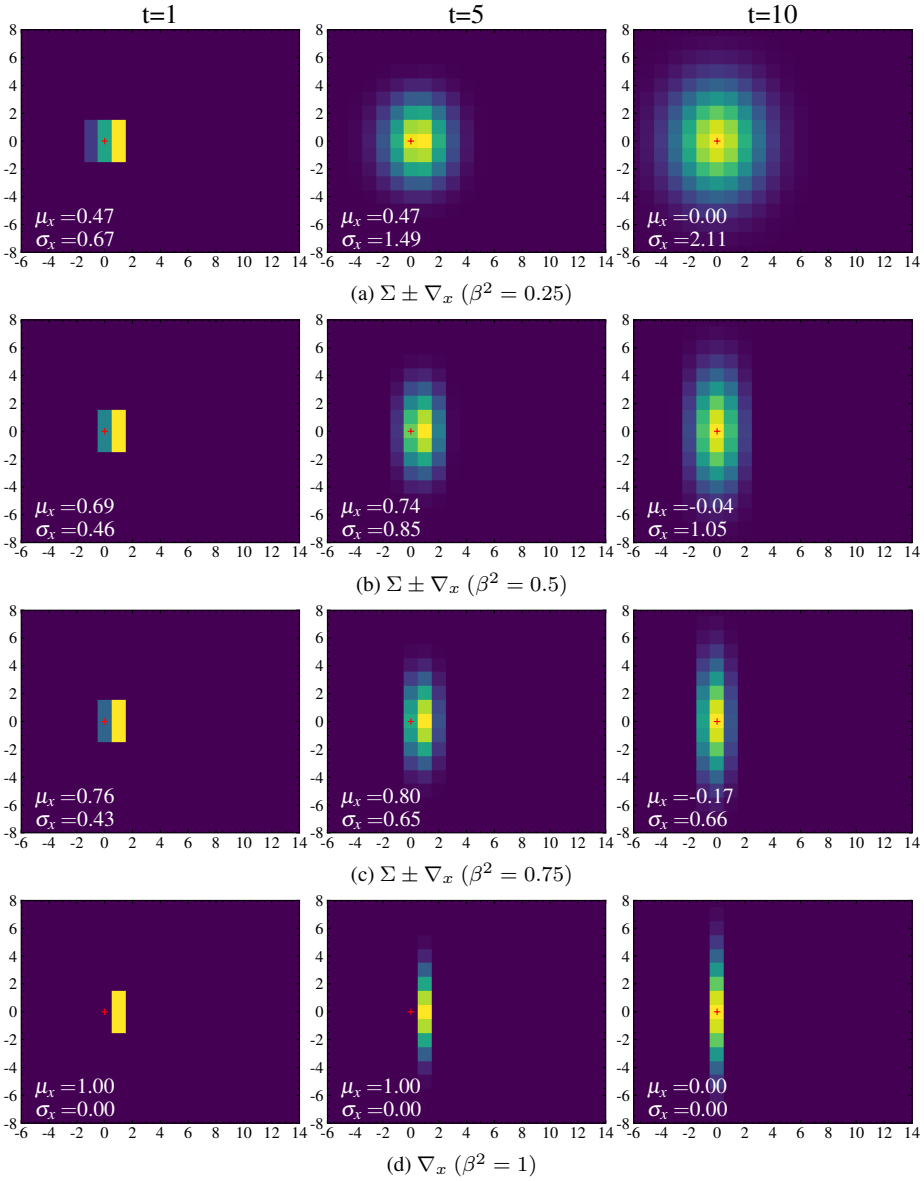
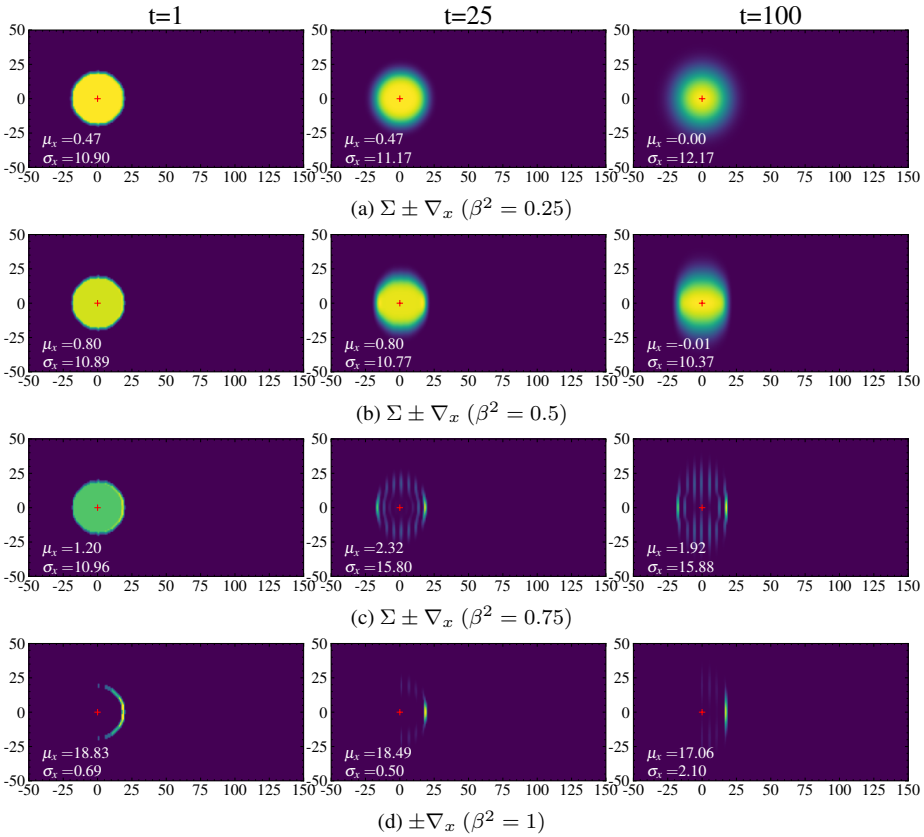


Figure 22. Demonstrating the effect of repeated convolution+ReLU with alternating orientation. Note that for $\beta^1 = 1$ in d), information vibrates left and right, and there is no net translation of the centre of mass.



(a) $\Sigma \pm \nabla_x$ ($\beta^2 = 0.25$)

(b) $\Sigma \pm \nabla_x$ ($\beta^2 = 0.5$)

(c) $\Sigma \pm \nabla_x$ ($\beta^2 = 0.75$)

(d) $\pm\nabla_x$ ($\beta^2 = 1$)

Figure 23. Demonstrating the effect of repeated convolution+ReLU of a circular test pattern over different types of 3×3 kernels mixing DC Σ and alternating direction gradient $\pm\nabla_x$ components for different mixing ratios. Note that for $\beta^2 = 1$, the circle bulk disappears, and the right edge of the circle vibrates left to right with no net translation.

C.3.4. 2×2 kernel, mixing unidirectional gradient ∇_x and sum Σ components

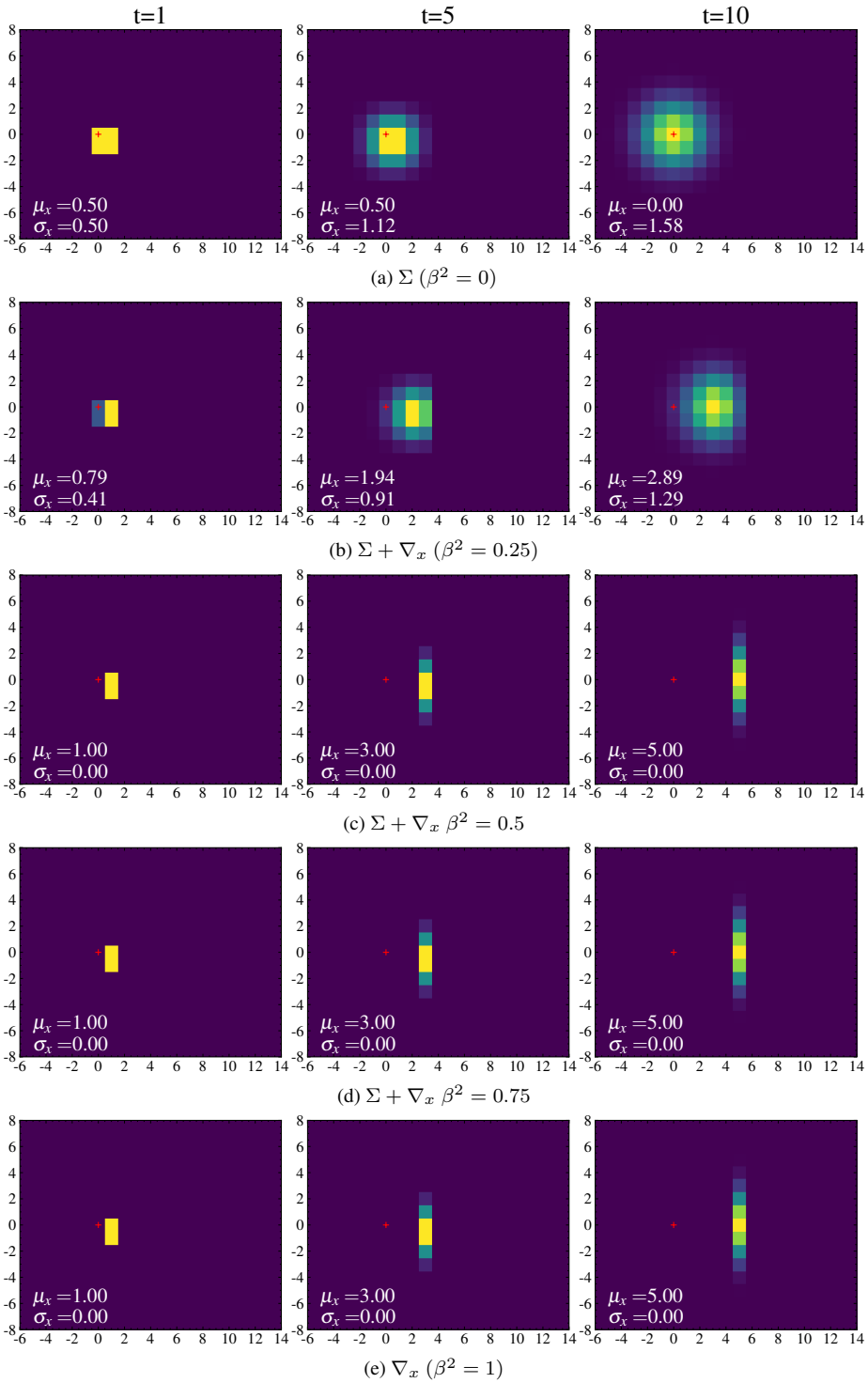


Figure 24. Demonstrating the effect of repeated convolution+ReLU of a test pattern over different types of 2×2 kernels (DC and Gradient). Note that for $\beta^2 \geq 0.5$, the content centre of mass travels rightward with maximum velocity.

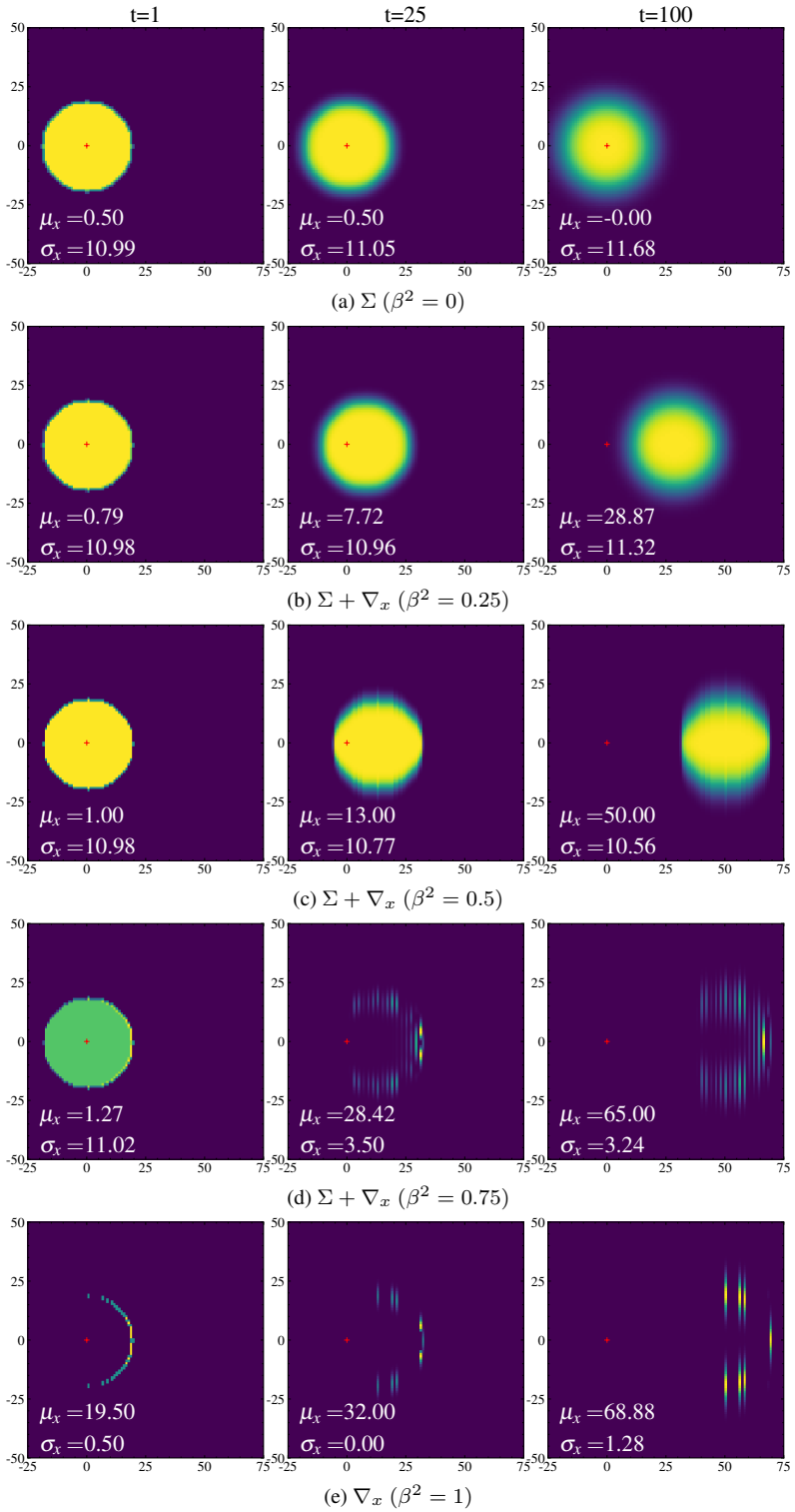
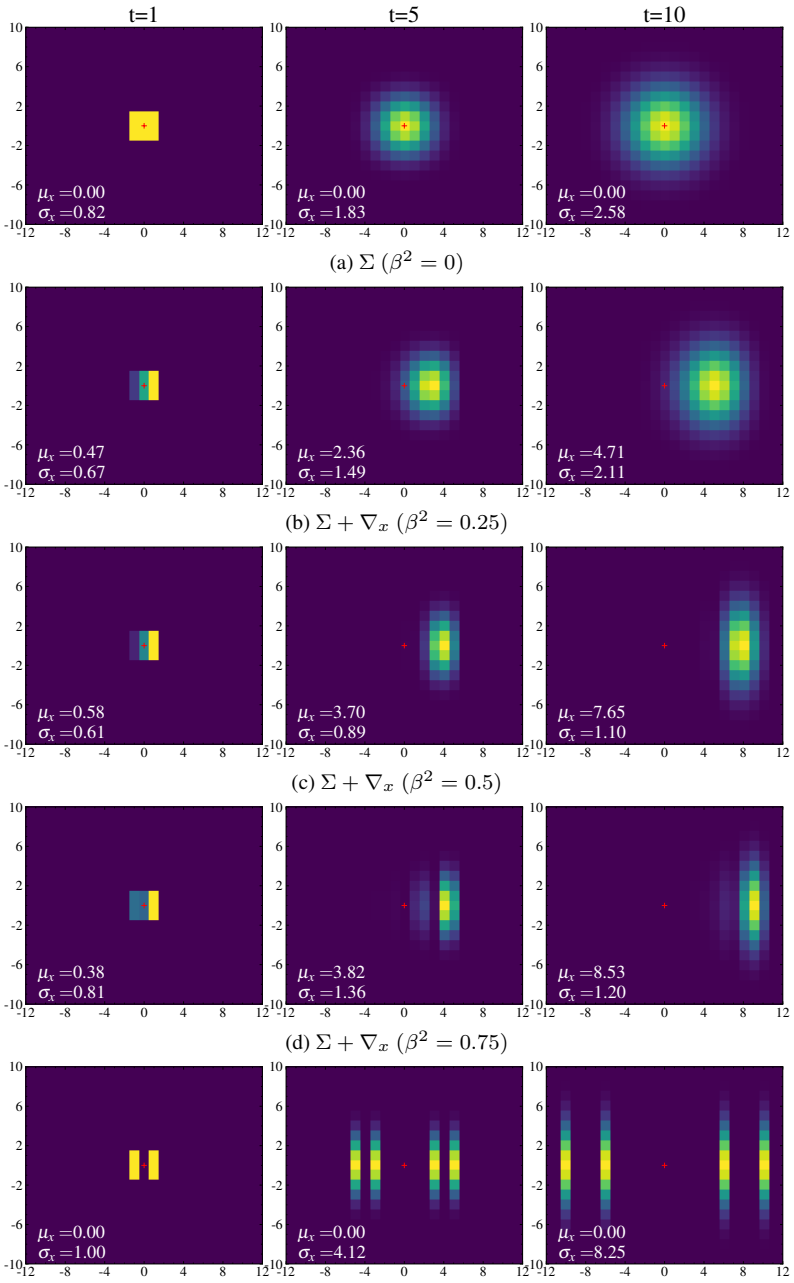


Figure 25. Demonstrating the effect of repeated convolution+ReLU of a circular test pattern ($r = 19$) with 2×2 kernels mixing DC Σ and fixed direction gradient ∇_x components over various mixing ratios β. Note that for $\beta = 0$, the content diffuses symmetrically with a stationary centre of mass, while for $\beta = 1$ the circle bulk disappears and the right edge of the circle travels rightward with maximum velocity.

C.3.5. 3×3 **kernel mixing unidirectional gradient** ∇_x **and sum** Σ **components, Modulus (Absolute value) activation.**



(a) Σ ($\beta^2 = 0$)

(b) $\Sigma + \nabla_x$ ($\beta^2 = 0.25$)

(c) $\Sigma + \nabla_x$ ($\beta^2 = 0.5$)

(d) $\Sigma + \nabla_x$ ($\beta^2 = 0.75$)

(e) ∇_x ($\beta^2 = 1$)

Figure 26. Demonstrating the effect of repeated **convolution+mod** of a test pattern over different types of 3×3 kernels (DC and Gradient). Note that for $\beta = 0$, the content diffuses symmetrically with a stationary centre of mass, while for $\beta = 1$ the pattern propagates symmetrically in both directions at maximum velocity with a stationary centre of mass.