Supplementary Materials for Robust Neural Networks inspired by Strong Stability Preserving Runge-Kutta methods

Byungjoo Kim[1⋆], Bryce Chudomelka[2⋆], Jinyoung Park[1],
Jaewoo Kang[1†], Youngjoon Hong[2], and Hyunwoo J. Kim[1†]

[1] Department of Computer Science, Korea University, Seoul, Republic of Korea
{byung4329,lpmn678,kangj,hyunwoojkim}@korea.ac.kr
[2] Department of Mathematics and Statistics, San Diego State University, San Diego, California, USA
{bchudomelka,yhong2}@sdsu.edu

1 Summary

This supplementary material is structured as follows: proofs of strong stability preserving methods (section 2), proofs of variance analysis of SSP networks (section 3), comparison of non-TVD scheme and TVD scheme (section 4), reminder of adversarial training (section 5), exploratory network analysis (section 6) and suppression on perturbation growth (section 7).

2 Proofs of Strong Stability Preserving Scheme

Lemma 1. *[5] If the forward Euler method is strongly stable under the CFL condition, i.e.* $||u^n + \Delta t L(u^n)|| \leq ||u^n||$*, then the Runge-Kutta method possesses SSP,* $||u^{n+1}|| \leq ||u^n||$*, provided that* $\Delta t \leq c\Delta t_{CFL}$*.*

Sketch of proof. To begin, we rewrite the Runge-Kutta method as a convex combination of forward Euler steps

$$\begin{aligned}\|u^{(i)}\| &= \left\| \sum_{k=0}^{i-1} \left(\alpha_{i,k} u^{(k)} + \Delta t \beta_{i,k} L(u^{(k)}) \right) \right\| \\ &\leq \sum_{k=0}^{i-1} \alpha_{i,j} \left\| u^{(k)} + \Delta t \frac{\beta_{i,k}}{\alpha_{i,k}} L(u^{(k)}) \right\|.\end{aligned}$$

If we set $c = \min_{i,k}(\alpha_{i,k}/\beta_{i,k})$ for $\Delta t \leq c\Delta t_{CFL}$, we find that

$$\|u^{(k)} + \Delta t \frac{\beta_{i,k}}{\alpha_{i,k}} L(u^{(k)})\| \leq \|u^{(k)}\|.$$

⋆ Equal Contribution, † Corresponding Author.

Also, we notice that $\sum_{k=0}^{i-1} \alpha_{i,k} = 1$ by consistency. We now use induction to show

$$\|u^{(k)}\| \leq \|u^n\|, \tag{1}$$

for $k = 0, 1, ..., m$. Clearly, when $k = 0$, (1) holds. Assuming that it is valid for all $k \leq i-1$, we deduce that

$$\begin{aligned}\|u^{(i)}\| &\leq \sum_{k=0}^{i-1} \alpha_{i,j} \left\| u^{(k)} + \Delta t \frac{\beta_{i,k}}{\alpha_{i,k}} L(u^{(k)}) \right\| \\ &\leq \sum_{k=0}^{i-1} \alpha_{i,k} \|u^{(k)}\| \\ &\leq \sum_{k=0}^{i-1} \alpha_{i,k} \|u^n\| = \|u^n\|.\end{aligned}$$

Hence, the lemma follows.

Lemma 2. *[5] An optimal second-order SSP Runge-Kutta method is given by,*

$$\begin{aligned}u^{(1)} &= u^n + \Delta t L(u^n), \\ u^{n+1} &= \frac{1}{2}u^n + \frac{1}{2}u^{(1)} + \frac{1}{2}\Delta t L(u^{(1)}),\end{aligned} \tag{2}$$

with a CFL coefficient $c = 1$. In addition, an optimal third-order SSP Runge-Kutta method is of the form

$$\begin{aligned}u^{(1)} &= u^n + \Delta t L(u^n), \\ u^{(2)} &= \frac{3}{4}u^n + \frac{1}{4}u^{(1)} + \frac{1}{4}\Delta t L(u^{(1)}), \\ u^{n+1} &= \frac{1}{3}u^n + \frac{2}{3}u^{(2)} + \frac{2}{3}\Delta t L(u^{(2)}),\end{aligned} \tag{3}$$

with a CFL coefficient $c = 1$.

Sketch of proof. For the second order $m = 2$, we choose the coefficients as

$$\begin{cases}\alpha_{1,0} = 1, \\ \alpha_{2,0} = 1 - \alpha_{2,1}, \\ \beta_{2,0} = 1 - \frac{1}{2\beta_{1,0}} - \alpha_{2,1}\beta_{1,0}, \\ \beta_{2,1} = \frac{1}{2\beta_{1,0}},\end{cases}$$

where $\beta_{1,0}$ and $\alpha_{2,1}$ are free parameters. Assume a CFL coefficient $c > 1$, then $\alpha_{1,0} = 1$ implies $\beta_{1,0} < 1$. Hence, we deduce that

$$\frac{1}{2\beta_{1,0}} > \frac{1}{2}.$$

In addition, we note that

$$\alpha_{2,1} > \beta_{2,1} = \frac{1}{2\beta_{1,0}} \implies \alpha_{2,1}\beta_{1,0} > \frac{1}{2}.$$

Hence, we obtain that

$$\beta_{2,0} = 1 - \frac{1}{2\beta_{1,0}} - \alpha_{2,1}\beta_{1,0} < 1 - \frac{1}{2} - \frac{1}{2} = 0,$$

which is a contradiction. For the third order case $m = 3$, we choose the coefficients as

$$\begin{cases} \alpha_{3,2} = 1 - \alpha_{3,1} - \alpha_{3,0}, \\ \beta_{3,2} = \dfrac{3\beta_{1,0} - 2}{6P(\beta_{1,0} - P)}, \\ \beta_{2,1} = \dfrac{1}{6\beta_{1,0}\beta_{3,2}}, \\ \beta_{3,1} = \dfrac{1/2 - \alpha_{3,2}\beta_{1,0}\beta_{2,1} - P\beta_{3,2}}{\beta_{1,0}}, \\ \beta_{3,0} = 1 - \alpha_{3,1}\beta_{1,0} - \alpha_{3,2}P - \beta_{3,1} - \beta_{3,2}, \\ \beta_{2,0} = P - \alpha_{2,1}\beta_{1,0} - \beta_{2,1}, \end{cases}$$

where $\alpha_{2,1}, \alpha_{3,0}, \alpha_{3,1}, \beta_{1,0}$, and $P = \beta_{2,0} + \alpha_{2,1}\beta_{1,0} + \beta_{2,1}$ are free parameters. We omit the detailed proof for the third order scheme as it is more technical. For the complete proof, see e.g. [1].

3 Proofs of Variance

In this section, we provide a proof of the Lemma 3.

Lemma 3. *If* $\mathtt{Var}[F(x)] = \mathtt{Var}[x]$, $\mathtt{Cov}[x, F(y)] = 0$ *then the variance increases by*

$$\begin{aligned} &\mathtt{Var}[\mathit{ResBlock(x)}] = 2\mathtt{Var}[x], \qquad &&\mathtt{Var}[\mathit{mid\text{-}RK2(x)}] = \frac{9}{4}\mathtt{Var}[x], \\ &\mathtt{Var}[\mathit{SSP2\text{-}Block(x)}] = \frac{7}{4}\mathtt{Var}[x], \qquad &&\mathtt{Var}[\mathit{SSP3\text{-}Block(x)}] = \frac{29}{18}\mathtt{Var}[x]. \end{aligned} \tag{4}$$

Proof To begin, we summarize basic properties of variance and convariance which commonly used in this proof.

$$\begin{aligned} &\mathtt{Var}[x + y] = \mathtt{Var}[x] + \mathtt{Var}[y] + 2\mathtt{Cov}[x, y], \\ &\mathtt{Var}[ax] = a^2\mathtt{Var}[x], \end{aligned} \tag{5}$$

$$\begin{aligned} &\mathtt{Cov}[x, y + z] = \mathtt{Cov}[x, y] + \mathtt{Cov}[x, z], \\ &\mathtt{Cov}[ax, by] = ab\mathtt{Cov}[x, y], \end{aligned} \tag{6}$$

where a, b are real-valued constants, x, y, z are random variables.
Our assumption holds

$$\texttt{Var}[F(x)] = \texttt{Var}[x] \tag{7}$$

and

$$\texttt{Cov}[x, F(y; \Theta)] = 0, \tag{8}$$

where x and y are random variables [2, 9]. Recall that operations of each block is written as

$$x_{k+1} = x_k + F(x_k; \Theta_k), \tag{9}$$

$$\begin{aligned} x_{k+\frac{1}{2}} &= x_k + \frac{1}{2}F(x_k; \Theta_k), \\ x_{k+1} &= x_k + F(x_{k+\frac{1}{2}}; \Theta_k), \end{aligned} \tag{10}$$

$$\begin{aligned} x_{k+\frac{1}{2}} &= x_k + F(x_k; \Theta_k), \\ x_{k+1} &= \frac{1}{2}x_k + \frac{1}{2}x_{k+\frac{1}{2}} + \frac{1}{2}F(x_{k+\frac{1}{2}}; \Theta_k), \end{aligned} \tag{11}$$

$$\begin{aligned} x_{k+\frac{1}{3}} &= x_k + F(x_k; \Theta_k), \\ x_{k+\frac{2}{3}} &= \frac{3}{4}x_k + \frac{1}{4}x_{k+\frac{1}{3}} + \frac{1}{4}F(x_{k+\frac{1}{3}}; \Theta_k), \\ x_{k+1} &= \frac{1}{3}x_k + \frac{2}{3}x_{k+\frac{2}{3}} + \frac{2}{3}F(x_{k+\frac{2}{3}}; \Theta_k), \end{aligned} \tag{12}$$

where the Equation (9) is the equation of *ResBlock*, Equation (10) is *mid-RK2 block*, Equation (11) is *SSP2-block*, Equation (12) is *SSP3-block*. We divide the proofs of each block. In every proofs, for simplicity, $F(x; \Theta_k) := f(x)$.
Proof of *ResBlock* Using (7) and (8), we derive the variance of output [9].

$$\begin{aligned} \texttt{Var}[x_{k+1}] &\overset{(5)}{=} \texttt{Var}[x_k] + \texttt{Var}[f(x_k)] + 2\texttt{Cov}[x_k, f(x_k)] \\ &\overset{(8)}{=} \texttt{Var}[x_k] + \texttt{Var}[f(x_k)] \\ &\overset{(7)}{=} 2\texttt{Var}[x_k]. \end{aligned}$$

Proof of *mid-RK2* First, we derive $\texttt{Var}[x_{k+\frac{1}{2}}]$ and $\texttt{Cov}[x_k, x_{k+\frac{1}{2}}]$.

$$\begin{aligned} \texttt{Var}[x_{k+\frac{1}{2}}] &\overset{(5)}{=} \texttt{Var}[x_k] + \frac{1}{4}\texttt{Var}[f(x_k)] + \texttt{Cov}[x_k, f(x_k)] \\ &\overset{(7,8)}{=} \frac{5}{4}\texttt{Var}[x_k], \end{aligned} \tag{13}$$

$$\begin{aligned}\mathtt{Cov}[x_k, x_{k+\frac{1}{2}}] &= \mathtt{Cov}[x_k, x_k + \frac{1}{2}f(x_k)] \\ &\overset{(6)}{=} \mathtt{Cov}[x_k, x_k] + \frac{1}{2}\mathtt{Cov}[x_k, f(x_k)] \\ &\overset{(7,8)}{=} \mathtt{Var}[x_k].\end{aligned} \tag{14}$$

By using $x_{k+1} = x_k + f(x_{k+\frac{1}{2}})$,

$$\begin{aligned}\mathtt{Var}[x_{k+1}] &\overset{(10)}{=} \mathtt{Var}[x_k + f(x_{k+\frac{1}{2}})] \\ &\overset{(5)}{=} \mathtt{Var}[x_k] + \mathtt{Var}[f(x_{k+\frac{1}{2}})] + 2\mathtt{Cov}[x_k, f(x_{k+\frac{1}{2}})] \\ &\overset{(7,8)}{=} \mathtt{Var}[x_k] + \mathtt{Var}[x_{k+\frac{1}{2}}] \\ &\overset{(13)}{=} \frac{9}{4}\mathtt{Var}[x_k].\end{aligned}$$

Proof of *SSP2-block* We start with obtaining $\mathtt{Var}[x_{k+\frac{1}{2}}]$ and $\mathtt{Cov}[x_k, x_{k+\frac{1}{2}}]$.

$$\begin{aligned}\mathtt{Var}[x_{k+\frac{1}{2}}] &\overset{(5)}{=} \mathtt{Var}[x_k] + \mathtt{Var}[f(x_k)] + 2\mathtt{Cov}[x_k, f(x_k)] \\ &\overset{(7,8)}{=} 2\mathtt{Var}[x_k],\end{aligned} \tag{15}$$

$$\begin{aligned}\mathtt{Cov}[x_k, x_{k+\frac{1}{2}}] &\overset{(6)}{=} \mathtt{Cov}[x_k, x_k] + \mathtt{Cov}[x_k, f(x_k)] \\ &\overset{(8)}{=} \mathtt{Cov}[x_k, x_k] \\ &= \mathtt{Var}[x_k].\end{aligned} \tag{16}$$

Next, let $x^{(1)} = x_{k+\frac{1}{2}} + f(x_{k+\frac{1}{2}})$. Then we can derive $\mathtt{Var}[x^{(1)}]$ and $\mathtt{Cov}[x_k, x^{(1)}]$.

$$\begin{aligned}&\mathtt{Var}[x^{(1)}] \\ &\overset{(5)}{=} \mathtt{Var}[x_{k+\frac{1}{2}}] + \mathtt{Var}[f(x_{k+\frac{1}{2}})] + 2\mathtt{Cov}[x_{k+\frac{1}{2}}, f(x_{k+\frac{1}{2}})] \\ &\overset{(8)}{=} \mathtt{Var}[x_{k+\frac{1}{2}}] + \mathtt{Var}[f(x_{k+\frac{1}{2}})] \\ &\overset{(7)}{=} \mathtt{Var}[x_{k+\frac{1}{2}}] + \mathtt{Var}[x_{k+\frac{1}{2}}] \\ &= 2\mathtt{Var}[x_{k+\frac{1}{2}}] \\ &\overset{(15)}{=} 4\mathtt{Var}[x_k],\end{aligned} \tag{17}$$

$$\begin{aligned}\mathtt{Cov}[x_k, x^{(1)}] &\overset{(6)}{=} \mathtt{Cov}[x_k, x_{k+\frac{1}{2}}] + \mathtt{Cov}[x_k, f(x_{k+\frac{1}{2}})] \\ &\overset{(8)}{=} \mathtt{Cov}[x_k, x_{k+\frac{1}{2}}] \\ &\overset{(16)}{=} \mathtt{Var}[x_k].\end{aligned} \tag{18}$$

Finally, by using $x^{(1)} = x_{k+\frac{1}{2}} + f(x_{k+\frac{1}{2}})$,

$$\begin{aligned}
\texttt{Var}[x_{k+1}] &\overset{(11)}{=} \texttt{Var}\left[\frac{1}{2}x_k + \frac{1}{2}x_{k+\frac{1}{2}} + \frac{1}{2}f(x_{k+\frac{1}{2}})\right] \\
&= \texttt{Var}\left[\frac{1}{2}x_k + \frac{1}{2}x^{(1)}\right] \\
&\overset{(5)}{=} \frac{1}{4}\texttt{Var}[x_k] + \frac{1}{4}\texttt{Var}[x^{(1)}] + \frac{1}{2}\texttt{Cov}[x_k, x^{(1)}] \\
&\overset{(17,18)}{=} \frac{1}{4}\texttt{Var}[x_k] + \texttt{Var}[x_k] + \frac{1}{2}\texttt{Var}[x_k] \\
&= \frac{7}{4}\texttt{Var}[x_k].
\end{aligned}$$

Proof of *SSP3-block* Similar to prove the *SSP2-block*, the first step is inducing $\texttt{Var}[x_{k+\frac{1}{3}}]$ and $\texttt{Cov}[x_k, x_{k+\frac{1}{3}}]$.

$$\texttt{Var}[x_{k+\frac{1}{3}}] = 2\texttt{Var}[x_k], \tag{19}$$

$$\begin{aligned}
\texttt{Cov}[x_k, x_{k+\frac{1}{3}}] &\overset{(6)}{=} \texttt{Cov}[x_k, x_k] + \texttt{Cov}[x_k, f(x_k)] \\
&\overset{(7)}{=} \texttt{Var}[x_k].
\end{aligned} \tag{20}$$

Let $x^{(1)} = x_{k+\frac{1}{3}} + f(x_{k+\frac{1}{3}})$. Then,

$$\begin{aligned}
&\texttt{Var}[x^{(1)}] \\
&\overset{(5)}{=} \texttt{Var}[x_{k+\frac{1}{3}}] + \texttt{Var}[f(x_{x+\frac{1}{3}})] + 2\texttt{Cov}[x_{k+\frac{1}{3}}, f(x_{x+\frac{1}{3}})] \\
&\overset{(8)}{=} \texttt{Var}[x_{k+\frac{1}{3}}] + \texttt{Var}[f(x_{x+\frac{1}{3}})] \\
&\overset{(7)}{=} 2\texttt{Var}[x_{k+\frac{1}{3}}] \\
&\overset{(19)}{=} 4\texttt{Var}[x_k],
\end{aligned} \tag{21}$$

$$\begin{aligned}
\texttt{Cov}[x_k, x^{(1)}] &\overset{(6)}{=} \texttt{Cov}[x_k, x_{k+\frac{1}{3}}] + \texttt{Cov}[x_k, f(x_{x+\frac{1}{3}})] \\
&\overset{(8,20)}{=} \texttt{Var}[x_k].
\end{aligned} \tag{22}$$

By using $x^{(1)} = x_{k+\frac{1}{3}} + f(x_{k+\frac{1}{3}})$,

$$\begin{aligned}
\texttt{Var}[x_{k+\frac{2}{3}}] &\overset{(12)}{=} \texttt{Var}\left[\frac{3}{4}x_k + \frac{1}{4}x_{k+\frac{1}{3}} + \frac{1}{4}f(x_{k+\frac{1}{3}})\right] \\
&= \texttt{Var}\left[\frac{3}{4}x_k + \frac{1}{4}x^{(1)}\right] \\
&\overset{(5)}{=} \frac{9}{16}\texttt{Var}[x_k] + \frac{1}{16}\texttt{Var}[x^{(1)}] + \frac{3}{8}\texttt{Cov}[x_k, x^{(1)}] \\
&\overset{(21,22)}{=} \frac{19}{16}\texttt{Var}[x_k],
\end{aligned} \tag{23}$$

and

$$\begin{aligned}\texttt{Cov}[x_k, x_{k+\frac{2}{3}}] &= \texttt{Cov}\left[x_k, \frac{3}{4}x_k + \frac{1}{4}x^{(1)}\right] \\ &\overset{(6)}{=} \frac{3}{4}\texttt{Cov}[x_k, x_k] + \frac{1}{4}\texttt{Cov}[x_k, x^{(1)}] \\ &\overset{(22)}{=} \texttt{Var}[x_k].\end{aligned} \tag{24}$$

Similar to previous steps, let $x^{(2)} = x_{k+\frac{2}{3}} + f(x_{k+\frac{2}{3}})$. Once again, by applying same procedure,

$$\begin{aligned}&\texttt{Var}[x^{(2)}] \\ &\overset{(5)}{=} \texttt{Var}[x_{k+\frac{2}{3}}] + \texttt{Var}[f(x_{k+\frac{2}{3}})] + 2\texttt{Cov}[x_{k+\frac{2}{3}}, f(x_{k+\frac{2}{3}})] \\ &\overset{(8)}{=} \texttt{Var}[x_{k+\frac{2}{3}}] + \texttt{Var}[f(x_{k+\frac{2}{3}})] \\ &\overset{(7)}{=} 2\texttt{Var}[x_{k+\frac{2}{3}}] \\ &\overset{(23)}{=} \frac{19}{8}\texttt{Var}[x_k],\end{aligned} \tag{25}$$

and

$$\begin{aligned}\texttt{Cov}[x_k, x^{(2)}] &\overset{(6)}{=} \texttt{Cov}[x_k, x_{k+\frac{2}{3}}] + \texttt{Cov}[x_k, f(x_{k+\frac{2}{3}})] \\ &\overset{(8)}{=} \texttt{Cov}[x_k, x_{k+\frac{2}{3}}] \\ &\overset{(24)}{=} \texttt{Var}[x_k].\end{aligned} \tag{26}$$

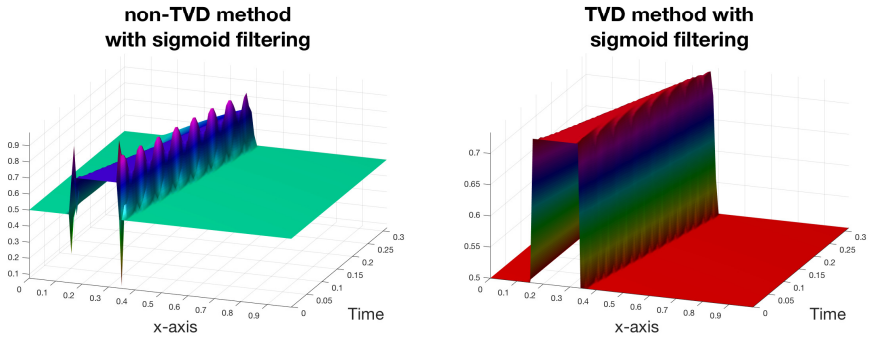
Finally, since $x^{(2)} = x_{k+\frac{2}{3}} + f(x_{k+\frac{2}{3}})$,

$$\begin{aligned}\texttt{Var}[x_{k+1}] &\overset{(12)}{=} \texttt{Var}\left[\frac{1}{3}x_k + \frac{2}{3}x_{k+\frac{2}{3}} + \frac{2}{3}f(x_{k+\frac{2}{3}})\right] \\ &= \texttt{Var}\left[\frac{1}{3}x_k + \frac{2}{3}x^{(2)}\right] \\ &\overset{(5)}{=} \frac{1}{9}\texttt{Var}[x_k] + \frac{4}{9}\texttt{Var}[x^{(2)}] + \frac{4}{9}\texttt{Cov}[x_k, x^{(2)}] \\ &\overset{(25,26)}{=} \frac{1}{9}\texttt{Var}[x_k] + \frac{19}{18}\texttt{Var}[x_k] + \frac{4}{9}\texttt{Var}[x_k] \\ &= \frac{29}{18}\texttt{Var}[x_k].\end{aligned}$$

4 Comparison non-TVD scheme and TVD scheme

In Figure 1, we implemented numerical solutions of the inviscid Burgers' equations

$$\begin{aligned}&u_t + uu_x = 0, \quad x \in (0,1), \\ &u(0,t) = u(1,t), \\ &u(x,0) = u_0,\end{aligned} \tag{27}$$



(a) non-TVD scheme (b) TVD scheme

Fig. 1: Two numerical solutions of the inviscid Burgers' equations using two different time discretizations are presented above. (a) shows the numerical solution with the non-TVD scheme in (28) while (b) is the numerical solution with the SSP3 discretization as in (3). After computing numerical solutions of the Burgers' equations, the solutions are filtered through the sigmoid function as an activation function. Evidently, the left panel (a) displays wild oscillations while the right panel (b) displays accurate numerical solutions.

using two different time discretizations; non-TVD scheme in (a) and TVD scheme in (b). The initial condition $u_0(x)$ is

$$u_0(x) := \begin{cases} 0, & 0 < x \leq 1/6, \\ 1, & 1/6 < x \leq 2/6, \\ 0, & 2/6 < x < 1. \end{cases}$$

The same equations and initial conditions were also used in Figure 1 of the main manuscript. For the spatial discretization, we adopt the third-order weighted essentially non-oscillatory (WENO) schemes. For numerical computations, the following configurations are used:

$$\begin{aligned} N &= \text{number of grid points in } x = 100, \\ h &= \text{Time step size} = 0.8/N, \\ T &= \text{final time} = 0.3. \end{aligned}$$

The left panel (a) shows the numerical solution with non-TVD time discretization of the second order while the right panel (b) presents the numerical solution with the SSP-2 discretization stated in (2). More precisely, in the left panel, we used the second order non-TVD scheme

$$\begin{aligned} u^{(1)} &= u^n - 20\Delta t L(u^n), \\ u^{n+1} &= u^n + \frac{41}{40}\Delta t L(u^n) - \frac{1}{40}\Delta t L(u^{(1)}). \end{aligned} \tag{28}$$

Algorithm 1 Projected Gradient Descent

Input: Clean image $x_{\text{nat}} \in [0,1]^m$ and corresponding label y, model h_θ, loss function ℓ, step α, bound ϵ, # of iteration K, metric p.
Output: Candidate adversarial example x_{adv}.

$x_{\text{adv}} := x_{\text{nat}}$
`feasible-set` $= \{x' | \|x' - x_{\text{nat}}\|_p \leq \epsilon\} \cap [0,1]^m$
for i **in** range(K) **do**
  $x_{\text{adv}} = x_{\text{adv}} + \alpha \cdot \texttt{sign}(\nabla_{x_{\text{adv}}} \ell(h_\theta(x_{\text{adv}}), y))$
  $x_{\text{adv}} = \Pi(x_{\text{adv}}, \texttt{feasible-set})$
end for
return x_{adv}

Algorithm 2 PGD Adversarial Training

Input: Training data minibatch (x_i, y_i) with $i \in \{1, \cdots, N\}$, initialized model h_θ, training epochs K, PGD attack algorithm `PGD`, Optimization method `optim`.
Output: Trained model h_θ

for i **in** range(K) **do**
  for j **in** range(N) **do**
    Sample random $\delta \in Uniform(-\epsilon, \epsilon)$
    $x_{j,\text{adv}} = x_{j,\text{nat}} + \delta$
    $x_{j,\text{adv}} = \texttt{PGD}(x_{j,\text{adv}}, x_{j,\text{nat}}, y_j)$
    $\theta := \texttt{optim}(h_\theta(x_{j,\text{adv}}), y_j)$
  end for
end for
return h_θ

After computing numerical solutions of the Burgers' equations, the solutions are filtered through the sigmoid function as activation function.

5 Adversarial Training Details

In this section, we briefly introduce the adversarial training which we used in our experiments.

Adversarial training is state-of-the-art methodology for defending adversarial attacks [4, 6, 8]. As we mentioned in our main paper, the objective of adversarial training is to minimize the adversarial risk given as

$$R_{adv}(h_\theta) = \mathbb{E}_{(x,y)\sim D}\big[\max_{\delta \in \Delta} \mathcal{L}(h_\theta(x+\delta), y)\big], \tag{29}$$

where the all notations are the same as our main paper. Strictly speaking, the set Δ has finite number of elements, so the exact solutions exist which maximize the loss $\mathcal{L}(h_\theta(x), y)$. However, as we mentioned in our main paper, numerically solving this problem is intractable. Therefore, most of works estimate the $\max_{\delta \in \Delta} \mathcal{L}(h_\theta(x), y)$ by using PGD; see Algorithm 1. Further, the randomness is injected during adversarial training, and this may help the robustness [4, 8]. The description of adversarial training is shown in Algorithm 2.

ϵ	Natural	1	2	3	4	5	6	7	8
ResNet	**88.14**	83.00	77.22	70.49	64.46	58.45	52.50	47.21	42.29
SSP-2	87.59	83.04	77.61	71.77	66.09	60.27	54.68	49.08	44.73
SSP-3	87.51	**83.31**	**78.49**	**72.70**	**66.61**	**60.90**	**55.28**	**50.20**	**45.40**

Table 1: Network performance against FGSM adversarial attacks when trained with PGD training ($\alpha = 1/255$). SSP-3 is more robust than ResNet and SSP-2; approximately 3%.

ϵ	Natural	1	2	3	4	5	6	7	8
ResNet	**88.14**	82.74	76.00	67.69	59.33	51.01	43.25	36.96	31.31
SSP-2	87.59	82.90	76.87	70.06	62.59	54.83	47.35	41.00	35.02
SSP-3	87.51	**83.16**	**77.78**	**70.75**	**63.19**	**55.53**	**48.51**	**42.08**	**36.13**

Table 2: Network performance against PGD adversarial attacks when trained with PGD training ($\alpha = 1/255$). SSP-3 is more robust than ResNet and SSP-2; approximately 5%.

6 Exploratory Network Analysis

In this section, we provide more experimental data on the CIFAR-10 dataset; as well as other well known datasets: Fashion-MNIST and Tiny-Imagenet.

We have trained ResNet, SSP-2 and SSP-3 on the CIFAR10 dataset, with $N = 5,\ K = 7$, and PGD adversarial training with $\alpha = 1/255$; in order to gauge performance and robustness of our architecture. We were able to perform various attacks on the network using the PGD and FGSM methods. We believe that this can be optimized with higher order methods, specifically SSP. Intuitively speaking, we are expecting performance to increase as a result of using a more accurate approximation. Also, accuracy should increase as we increase the order of the numerical approximation.

We noticed that when all three models were attacked via FGSM, or PGD, with $\alpha = 1/255$, and various ϵ, that the higher order methods outperformed ResNet from roughly $3 \sim 5\%$. We observed that SSP-3 outperforms SSP-2, which outperforms ResNet; consistently. Furthermore, as the strength of the perturbation was increased, SSP networks became more resilient to adversarial attacks than ResNet. This results in behavior that is truly characteristic of numerical methods, in that accuracy is increased as higher order methods are implemented, and stability is preserved.

What is perhaps the most notable is that SSP-3 is extremely more robust than ResNet whether it is attacked via FGSM or PGD. Robustness is achieved without introducing more parameters, or a dramatic increase to computational

Model	Clean	FGSM	PGD_{20}
ResNet	0.9090	0.8562	0.8179
SSP-2	0.9132	0.8591	0.8252
SSP-3	0.9110	**0.8639**	**0.8295**
SSP-adap	0.9098	0.8621	0.8264

Table 3: Result on Fashion-MNIST. All the models follow the same architecture which used in MNIST experiment in main paper. The number of blocks is 20, with using Group Normalization. We perform adversarial training with $\epsilon = 0.1$, $\alpha = 0.02$ with 10 iterations. For evaluating robustness, $\alpha = 0.01$ with 20 iterations are used in PGD attack (PGD_{20}).

Model	Clean	PGD_{20}
ResNet	0.4648	0.1738
SSP-2	0.4386	0.1761
SSP-3	0.4529	**0.1955**

Table 4: Result on Tiny-Imagenet. We perform adversarial training with $\epsilon = 8/255$, $\alpha = 2/255$ with 5 iterations. For evaluating robustness, $\alpha = 2/255$ with 20 iterations are used in PGD attack (PGD_{20}).

power. Our architecture achieves comparable performance on unperturbed images and superior performance with respect to adversarial attacks.

Next, we evaluate the robustness on Fashion-MNIST [7] dataset. The architecture of model is same as the model used in MNIST, but the only difference is the number of blocks and maximum perturbation range (ϵ). The results are shown in Table 3 and are consistent with our previous assumptions and results.

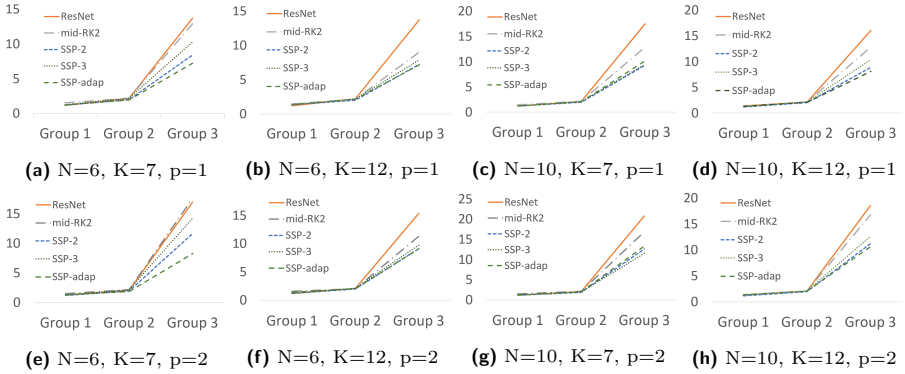
Last but not least, we conduct an experiment on the more challenging dataset, Tiny-Imagenet [3]. The models used in Tiny-Imagenet experiment are composed of 4 groups of blocks and each group has 10 blocks. Table 4 shows the top-1 accuracy of natural samples and adversarial examples generated by PGD attack. As the result shows, all the SSP networks show better robustness than ResNet.

7 Suppression on Perturbation Growth

In this section, we present the perturbation growth ratio of all the networks used in the CIFAR-10 experiments. Recall that the perturbation growth ratio is given by

$$\texttt{PGR}(f) = \mathbb{E}_{x\sim\mathcal{D}}\left[\mathbb{E}_{x'\sim\mathcal{X}'}\left[\frac{\|f(x)-f(x')\|_p}{\|x-x'\|_p}\right]\right], \quad p \in \{1,2\},$$

where each corrupted sample x' is sampled from a small neighborhood of x, i.e., $\mathcal{X}'$, and p defines a type of norm either ℓ_1 or ℓ_2.



(a) N=6, K=7, p=1 **(b)** N=6, K=12, p=1 **(c)** N=10, K=7, p=1 **(d)** N=10, K=12, p=1

(e) N=6, K=7, p=2 **(f)** N=6, K=12, p=2 **(g)** N=10, K=7, p=2 **(h)** N=10, K=12, p=2

Fig. 2: Perturbation growth ratio of clean samples and its adversarial counterparts. As the perturbation evolves through networks, SSPNets have lower ratio than ResNet.

In Figure 2, all the corrupted sample x' is the adversarial example generated by PGD attack with 20 iterations for each model. Despite there is no other regularization using Lipschitzness or Jacobian, all the SSPNets have lower perturbation growth ratio than ResNet.

References

1. Gottlieb, S., Shu, C.W.: Total variation diminishing runge-kutta schemes. Mathematics of computation of the American Mathematical Society **67**(221), 73–85 (1998)
2. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: Proceedings of the IEEE international conference on computer vision. pp. 1026–1034 (2015)
3. Le, Y., Yang, X.: Tiny imagenet visual recognition challenge. CS 231N (2015)
4. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. In: ICLR (2018)
5. Shu, C.W., Osher, S.: Efficient implementation of essentially non-oscillatory shock-capturing schemes. Journal of computational physics **77**(2), 439–471 (1988)
6. Wang, Y., Ma, X., Bailey, J., Yi, J., Zhou, B., Gu, Q.: On the convergence and robustness of adversarial training. In: ICML. pp. 6586–6595 (2019)
7. Xiao, H., Rasul, K., Vollgraf, R.: Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:1708.07747 (2017)
8. Xie, C., Wu, Y., Maaten, L.v.d., Yuille, A.L., He, K.: Feature denoising for improving adversarial robustness. In: CVPR. pp. 501–509 (2019)
9. Zhang, H., Dauphin, Y.N., Ma, T.: Residual learning without normalization via better initialization. In: International Conference on Learning Representations (2019)