A Iterative Procedure for the Inverse of i-ResNET

Let x_{t+1} in (1) be equal to x_t^0. Then, the inverse of F_t in (1), F_t^{-1}, is given by the following procedure:

$$\begin{aligned} x_t^0 - g_t\left(x_t^0\right) &= x_t^1 \\ x_t^0 - g_t\left(x_t^1\right) &= x_t^2 \\ &\vdots \\ x_t^0 - g_t\left(x_t^{N-1}\right) &= x_t^N, \end{aligned} \tag{89}$$

where $N \in \mathbb{N}$ is the total number of iteration steps. This iterative process is called the ***Banach fixed point iteration*** and as $N \to \infty$, $x_t^N \to x_t$. Once we keep substituting equivalent statements for x_t^i until the N^{th} iteration:

$$\begin{aligned} x_{t+1} - g_t\left(x_t^0\right) &= x_t^1 \\ x_{t+1} - g_t\left[x_{t+1} - g_t\left(x_{t+1}\right)\right] &= x_t^2 \\ \implies x_{t+1} - g_t\left[x_{t+1}\right] + g_t^2\left(x_{t+1}\right) &= x_t^2 \\ &\vdots \\ x_{t+1} + \left\{\sum_{i=1}^{N}(-1)^i g_t^i\right\}\left(x_{t+1}\right) &= x_t^N \\ \implies \left\{\sum_{i=0}^{N}(-1)^i g_t^i\right\}\left(x_{t+1}\right) &= x_t^N \end{aligned} \tag{90}$$

Therefore, as $N \to \infty$ in (90), the inverse, F_t^{-1} is obtained:

$$F_t^{-1}\left(x_{t+1}\right) = \lim_{N\to\infty}\left\{\sum_{i=0}^{N}(-1)^i g_t^i\right\} x_{t+1} = \lim_{N\to\infty} x_t^N = x_t \tag{91}$$

It is easy to show that (91) is actually the inverse of F_t. Consider the following two series:

$$\begin{aligned} I - g_t + g_t^2 - \cdots \mp g_t^N &= S_N \\ g_t - g_t^2 + \cdots \mp g_t^{N+1} &= g_t S_N, \end{aligned} \tag{92}$$

where $S_N = \sum_{i=0}^{N}(-1)^i g_t^i$. Adding up the left and right hand sides in (92), respectively, yields:

$$I \mp g_t^{N+1} = \left[I + g_t\right] S_N \tag{93}$$

Since $Lip\left(g_t\right) < 1$, $g_t^{N+1} \to 0$ transformation as $N \to \infty$. Thus, (93) becomes:

$$I = \left[I + g_t\right] \lim_{N\to\infty} S_N \tag{94}$$

By (94), we conclude that $\lim_{N\to\infty} S_N$ is the inverse of F_t.

B Right Invariant Vector Fields & Vanishing Lie-Bracket

Let G be a connected Lie group equipped with Lie algebra $\mathfrak{g}$ and $\dim(\mathfrak{g}) = r \geq 2$. Let $\{v_1, \ldots, v_r\}$ span $\mathfrak{g}$ which is isomorphic to $TG|_e$ by **Corollary 3.46** in [18], where $[v_i, v_j] \in \mathfrak{g}$ is not vanishing for any i and j. Furthermore; $v_i \in \mathscr{R}(G), \forall\ i = 1, \ldots, r$, where $\mathscr{R}(G)$ is the space of right invariant vector fields on G. By **Frobenius' theorem** (see **Lemma F.2** and **Corollary F.3** in [19]), one can always conclude that $\exists\ \{B_1, \ldots, B_r\}$ that spans $\mathfrak{g}$ s.t. $[B_i, B_k] = 0, \forall i, k = 1, \ldots, r$ in a small neighbourhood, U, of e. However, this is not the case for right invariant vector fields in G. Following setup is required before proving why *Frobenius' theorem* fails:

(i) Let $R_g : G \to G$ be right translation, i.e. $R_g(h) = h.g$, where $h, g \in G$.

(ii) Suppose $\mathfrak{g} \cong \operatorname{span}\{v_1, \ldots, v_r\}_{|e}$ and $v_l \in \mathscr{R}(G), \forall\ l = 1, \ldots, r$ and $[v_i, v_k] \neq 0, \forall i \neq k$.

(iii) Assume also $\mathfrak{g} \cong \text{span}\{B_1, \ldots, B_r\}_{|_e}$, s.t. $[B_i, B_k] = 0$, $\forall\, i, k = 1, \ldots, r$ in a small neighbourhood, U, of e due to **Frobenius' theorem**.

(iv) Following the same notation in [7], let $v_l|_g = \sum_p \eta_l^p(g)\partial_p$, $\forall\, l = 1, \ldots, r$ and $\forall\, g \in G$.

(v) Since $\{v_1, \ldots, v_r\}$ spans $\mathfrak{g}$, $B_i = \sum_l \alpha_i^l(\varepsilon)\, v_l$, $\forall\, i, l = 1, \ldots, r$, where $\alpha_i : \mathbb{R} \to \mathbb{R}^r$ are smooth functions. We are going to omit α_i^l's arguments for brevity.

(vi) Since G is connected, by **Proposition 1.48** in [7] $Y = v_l$, $\forall\, l = 1, \ldots, r$ generates one-parameter subgroup s.t. $A(t) = exp(tY|_e)$.

(vii) Let $Y = v_q$ for a particular index q and g in item-(i) be equal to $exp(t_g Y|_e)$ for a particular small $t_g \in \mathbb{R} \setminus \{0\}$, s.t. $g \in G$.

Let B_i and B_k be our two candidate vector fields for a particular pair of $(i, k) \in \{1, \ldots, r\} \times \{1, \ldots, r\}$. The assumption is: ***"Suppose that*** $\mathbf{B_i}$ ***and*** $\mathbf{B_k} \in \mathcal{R}(\mathbf{G})$***"***. Thus, we should expect: $[dR_g(B_i|_e), dR_g(B_k|_e)] = [B_i|_g, B_k|_g] = 0$, where the first equality is due to (**1.32**) in [7] and dR_g is the differential for R_g and is equal to:

$$dR_g\left(B_i|_e\right) = \sum_j B_i\left(R_g^j(e)\right)\partial_j = \sum_j \sum_l \alpha_i^l v_l\left(R_g^j(e)\right)\partial_j = \sum_j \sum_l \sum_p \alpha_i^l \eta_l^p(e) \frac{\partial R_g^j}{\partial x^p}(e)\partial_j, \tag{95}$$

where $\{x^p\}_{p=1}^m$ are coordinates of the Euclidean space into which G is embedded and the first equality is due to (**1.23**) in [7]. Since $R_g(h) = h.exp(t_g Y|_e)$:

$$R_g(h) = R_h(e).\exp(t_g Y|_e) = \exp(t_g dR_h(Y|_e)) = \exp(t_g Y(R_h(e))) = \exp(t_g Y|_h) = h + t_g \sum_p \eta_q^p(h) + \theta(t_g^2), \tag{96}$$

where the second and third equality are again due to (**1.23**) in [7]. Therefore, the j^{th} component function of R_g is:

$$R_g^j(h) = x^j(h) + t_g \eta_q^j(h) + \theta^j(t_g^2), \tag{97}$$

where $x^j(h) = h^j$ and $\theta^j(.)$ include higher order terms. Therefore:

$$\frac{\partial R_g^j}{\partial x^p}(h) = \delta x_p^j(h) + t_g \frac{\partial \eta_q^j}{\partial x^p}(h) + \theta^j(t_g^2), \tag{98}$$

Substitute (98) into (95):

$$\begin{aligned}
&= \sum_j \sum_l \sum_p \alpha_i^l \eta_l^p(e)\left(\delta x_p^j(h) + t_g \frac{\partial \eta_q^j}{\partial x^p}(h) + \theta^j(t_g^2)\right)\partial_j \\
&= \sum_j \left(\sum_l \alpha_i^l \eta_l^j(e) + t_g \sum_l \sum_p \alpha_i^l \eta_l^p(e) \frac{\eta_q^j}{\partial x^p}(e) + \theta^j(t_g^2)\right)\partial_j \\
&= B_i|_e + t_g B_i(Y)|_e + \theta(t_g^2),
\end{aligned} \tag{99}$$

All equations from (95) to (99) apply to B_k as well. Therefore, we may define the Lie bracket between B_i and B_k as follows:

$$\begin{aligned}
[B_i|_g,\, B_k|_g] &= \left[dR_g\left(B_i|_e\right),\, dR_g\left(B_k|_e\right)\right] = [B_i|_e + t_g B_i(Y)|_e + \theta(t_g^2), B_k|_e + t_g B_k(Y)|_e + \theta(t_g^2)] \\
&= \underbrace{[B_i|_e,\, B_k|_e]}_{=0} + t_g\left[B_i|_e,\, B_k(Y)|_e\right] + t_g\left[B_i(Y)|_e,\, B_k|_e\right] + \underbrace{\theta(t_g^2)}_{\approx 0},
\end{aligned} \tag{100}$$

where the bilinearity property of Lie bracket is used to obtain the last equality. The first term is equal to 0 by our initial assumption; while the last term is approximately equal to 0 once we assume t_g is sufficiently small. However, the second and the third terms may not be equal to zero in (100). Consider the second term:

$$t_g\Bigg\{\sum_{l_1}\alpha_i^{l_1}\left(\sum_{p_1}\eta\,(e)_{l_1}^{p_1}\right)\partial_{p_1}\left(\sum_{l_2}\alpha_k^{l_2}\underbrace{\left(\sum_{p_2}\eta\,(e)_{l_2}^{p_2}\,\partial_{p_2}\left(\sum_{p_3}\eta\,(e)_q^{p_3}\right)\right)}_{\omega_{l_2}|_e=\sum_{p_3}\xi(e)_{l_2}^{p_3}\partial_{p_3}}\right)-$$
$$\left(\sum_{l_2}\alpha_k^{l_2}\underbrace{\left(\sum_{p_2}\eta\,(e)_{l_2}^{p_2}\,\partial_{p_2}\left(\sum_{p_3}\eta\,(e)_q^{p_3}\right)\right)}_{\omega_{l_2}|_e=\sum_{p_3}\xi(e)_{l_2}^{p_3}\partial_{p_3}}\right)\sum_{l_1}\alpha_i^{l_1}\left(\sum_{p_1}\eta\,(e)_{l_1}^{p_1}\right)\Bigg\} \tag{101}$$
$$=t_g\left[\sum_{l_1}\alpha_i^{l_1}\sum_{p_1}\eta\,(e)_{l_1}^{p_1},\sum_{l_2}\alpha_k^{l_2}\omega_{l_2}|_e\right]=t_g\left[\sum_{l_1}\alpha_i^{l_1}v_{l_1},\sum_{l_2}\alpha_k^{l_2}\omega_{l_2}\right]\Bigg|_e,$$

where $\omega_{l_2}\in\mathscr{R}(G)$, $\forall\, l_2=1,\ldots,r$, because v_{l_2}'s action on v_q leave v_q in $\mathfrak{g}$ and not all $\omega_{l_2}=0$ for $l_2=1,\ldots,r$. Similarly the third term in (100) is equal to:

$$t_g\left[\sum_{l_1}\alpha_i^{l_1}\omega_{l_1},\sum_{l_2}\alpha_k^{l_2}v_{l_2}\right]\Bigg|_e. \tag{102}$$

Adding (101) to (102):

$$\begin{aligned}
&t_g\left\{\left[\sum_{l_1}\alpha_i^{l_1}v_{l_1},\sum_{l_2}\alpha_k^{l_2}\omega_{l_2}\right]\Bigg|_e+\left[\sum_{l_1}\alpha_i^{l_1}\omega_{l_1},\sum_{l_2}\alpha_k^{l_2}v_{l_2}\right]\Bigg|_e\right\}=\\
&t_g\left\{\left[\sum_{l_1}\alpha_i^{l_1}v_{l_1},\sum_{l_2}\alpha_k^{l_2}\omega_{l_2}\right]\Bigg|_e+\left[\sum_{l_2}\alpha_i^{l_2}\omega_{l_2},\sum_{l_1}\alpha_k^{l_1}v_{l_1}\right]\Bigg|_e\right\}=\\
&t_g\left\{\left[\sum_{l_1}\alpha_i^{l_1}v_{l_1},\sum_{l_2}\alpha_k^{l_2}\omega_{l_2}\right]\Bigg|_e-\left[\sum_{l_1}\alpha_k^{l_1}v_{l_1},\sum_{l_2}\alpha_i^{l_2}\omega_{l_2}\right]\Bigg|_e\right\}=\\
&t_g\sum_{l_2}\left\{\left[\sum_{l_1}\alpha_k^{l_2}\alpha_i^{l_1}v_{l_1},\omega_{l_2}\right]\Bigg|_e-\left[\sum_{l_1}\alpha_i^{l_2}\alpha_k^{l_1}v_{l_1},\omega_{l_2}\right]\Bigg|_e\right\}=\\
&t_g\sum_{l_2}\Bigg[\underbrace{\sum_{l_1}\left(\alpha_k^{l_2}\alpha_i^{l_1}-\alpha_i^{l_2}\alpha_k^{l_1}\right)v_{l_1}}_{=\beta_{l_2}},\omega_{l_2}\Bigg]\Bigg|_e,
\end{aligned} \tag{103}$$

where bilinearity of Lie bracket is used for obtaining the last two equalities and $\beta_{l_2}\in\mathscr{R}(G)$, $\forall\, l_2=1,\ldots,r$, because the Lie algebra is closed under scalar multiplication and vector addition. Besides; $\beta_{l_2}|_e$ cannot be equal to the 0 vector for any $l_2=1,\ldots,r$, because it would have implied that $\alpha_k^{l_2}B_i=\alpha_i^{l_2}B_k$, meaning that B_i and B_k were linearly dependent violating our initial assumption. Thus, (100) cannot be equal to 0 resulting in our assumption that right invariant vector fields abide by **Frobenius' theorem** is false.

C Smooth Orthonormal Global Frames for Connected Lie Groups

Let G be a connected Lie group with corresponding Lie algebra $\mathfrak{g}\cong T\mathscr{M}|_e$, where $\mathscr{M}$ is its underlying manifold in respective representation space. By **Corollary 8.39** in [33], every Lie group G has a smooth global frame, $\{X_1,\ldots,X_r\}$, i.e. right (left) invariant basis at $T\mathscr{M}|_g$, $\forall\, g\in\mathscr{M}$ and component functions for X_j, $\forall\, j=1,\ldots,r$ are smooth. However, it does not mean there <u>*always*</u> exists an *orthonormal* smooth global frame for every Lie group. Frankly, there is none for any Lie group other than abelian ones. By **Theorem 13.4** in [33], having a global orthonormal coordinate frame implies having $\mathscr{M}$ contained in the domain of a commuting orthonormal frame $\forall\, g\in\mathscr{M}$, i.e. $[X_i\,,X_j]\,|_g=0$, $\forall\, i,j=1,\ldots,r$ and $\forall\, g\in\mathscr{M}$. As discussed in

Appendix B, $[X_i\,,X_j]\,|_g \neq 0$, $\forall\, i \neq j$ and $\forall\, g \in \mathscr{M}$. Thus, we may conclude that there exists no smooth global orthonormal frame for G provided that it is not abelian. In our experiments, it also comes as no surprise when *right invariance cost* in (45) and *consistency cost* in (53) are turned off, the optimization converges to a location in parameter space where orthogonality condition at every sample point is satisfied for a particular subgroup under training.

D Right Invariance of Vector Fields in $\mathfrak{g}$ and Lie Bracket

In this section, how ensuring right invariance yields a proper Lie bracket for $\mathfrak{g}$ is discussed. It is also significant in the sense of computational efficiency, because computing Lie-bracket would have been costlier than is computing right invariance cost $\mathscr{R}\mathscr{I}$ given in (45). Besides, guaranteeing right invariance for each $X_i \in \mathfrak{g}$ implies a well-defined Lie-bracket; but not vice-versa.

Right invariance of a vector field X_i in mathematical terms equal to the following as stated in (44):

$$d\mathscr{T}_{\exp\left(\varepsilon X_j\right)}\left[X_i|_p\right] = X_i|_{\exp\left(\varepsilon X_j\right).p},\ \forall\, i,j = 1,\ldots,r, \tag{104}$$

where $d\mathscr{T}_{\exp\left(\varepsilon X_j\right)}$ is the differential of group action $\mathscr{T}_j$. We tweak the LHS of (104) to suit our needs and let it be equal to $d\mathscr{T}_{\exp\left(\varepsilon X_j\right)}\left[X_i|_{\exp\left(-\varepsilon X_j\right).p}\right]$. In other words, our new smooth map is:

$$d\mathscr{T}_{\exp\left(\varepsilon X_j\right)}\left[.|_{\exp\left(-\varepsilon X_j\right).p}\right] : T\mathscr{N}|_{\exp\left(-\varepsilon X_j\right).p} \rightarrow T\mathscr{N}|_p. \tag{105}$$

Note that $d\mathscr{T}_{\exp\left(\varepsilon X_j\right)}\left[X_i|_{\exp\left(-\varepsilon X_j\right).p}\right] = X_i|_p$, as we assume that $X_i \in \mathfrak{g}$, $\forall\, i = 1,\ldots,r$ are right invariant vector fields over G. Moreover, since we assume $X_i \in \mathfrak{g}$ are smooth vector fields, the following Taylor approximations

$$d\mathscr{T}_{\exp\left(\varepsilon X_j\right)}|_p = \left[\delta_q^l + \varepsilon\frac{\partial\zeta_j^l}{\partial p^q} + \Theta\left(\varepsilon^2\right)\right]_q^l(p), \tag{106}$$

and

$$X_i|_{\exp\left(-\varepsilon X_j\right).p} = \left[\zeta_i^l - \varepsilon\sum_{k=1}^{m}\zeta_j^k\frac{\partial\zeta_i^l}{\partial p^k} + \Theta\left(\varepsilon^2\right)\right]_l(p) \tag{107}$$

are well-defined, where δ_q^l is the delta function and we assume that $X_{i,j}|_p = \left[\zeta_{i,j}^1(p),\ldots,\zeta_{i,j}^l(p),\ldots,\zeta_{i,j}^n(p)\right]^T$. Using (106) and (107), $d\mathscr{T}_{\exp\left(\varepsilon X_j\right)}\left[X_i|_{\exp\left(-\varepsilon X_j\right).p}\right]$ could be reformulated as:

$$\begin{aligned} &= \left[\zeta_i^q + \varepsilon\left(\sum_{k=1}^{m}\zeta_i^k\frac{\partial\zeta_j^k}{\partial p^k} - \zeta_j^k\frac{\partial\zeta_i^q}{\partial p^k}\right) + \Theta\left(\varepsilon^2\right)\right]_q(p) \\ &= \left[\zeta_i^q + \varepsilon\left[X_i,X_j\right]^q + \Theta\left(\varepsilon^2\right)\right]_q(p), \end{aligned} \tag{108}$$

where $[X_i,X_j]^q$ is the q^{th} component of the vector field generated by Lie bracket of X_i and X_j. Since $\mathfrak{g}$ is a vector space (which is closed under scalar multiplication and vector addition) and $\varepsilon > 0$, we may conclude that $[X_i,X_j] \in \mathfrak{g}$.

E The Proof of Proposition 2

Let $\mathscr{T}_j : \mathscr{P} \rightarrow \mathscr{P}$ as defined in (32) for a fixed $0 < \varepsilon^j \ll 1 \in \mathbb{R}$. To prove Proposition 2, $\left(X_j|_p\right)^n : \mathscr{P} \rightarrow \mathscr{P}, \forall n \geq 1$ should be treated as an *operator* acting on prolonged space $\mathscr{P}$. Let $p \in \mathscr{P}$ and $\left(X_j|_p\right)^0 \equiv p$. Then:

$$\left(X_j|_p\right)^1 = \sum_{i_1=1}^{r}\zeta^{i_1}(p)\,{}^{\partial}\!/\!_{\partial p^{i_1}}, \tag{109}$$

i.e. evaluation of vector field at that point. Defining the operator for $n = 2$ allows one to easily grasp the general iterative formula that will be given in (111):

$$\left(X_j|_p\right)^2 = \sum_{i_1=1}^{r}\left(\sum_{i_2=1}^{r}\zeta^{i_2}(p)\frac{\partial\zeta^{i_1}(p)}{\partial p^{i_2}}\right){}^{\partial}\!/\!_{\partial p^{i_1}}. \tag{110}$$

Using the same procedure employed both in (109) and (110) which treats $X_j = \sum_{i=1}^{r} \zeta^i(.)\,\partial/\partial p^i$ as an operator eating a position vector $p \in \mathscr{P}$ and spitting out a vector $v \in \mathbb{R}^m$, the $(X_j|_p)^n$ for any $n \geq 1$ becomes:

$$(X_j|_p)^n = \sum_{i_1=1}^{r} \left\{ \sum_{i_2=1}^{r} \zeta^{i_2}(p) \frac{\partial}{\partial p^{i_2}} \left\{ \cdots \sum_{i_{n-2}=1}^{r} \zeta^{i_{n-2}}(p) \frac{\partial}{\partial p^{i_{n-2}}} \left\{ \sum_{i_{n-1}=1}^{r} \zeta^{i_{n-1}}(p) \frac{\partial \zeta^{i_1}}{\partial p^{i_{n-1}}}(p) \right\} \cdots \right\} \right\} \partial/\partial p^{i_1}. \tag{111}$$

Now, suppose that we have two position vectors $p_1, p_2 \in \mathscr{P}$ that are sufficiently close w.r.t. Euclidean norm induced by inner product defined over $\mathbb{R}^m$. Observe that it is not necessary to define a metric on manifold, $\mathscr{N}$, for this proof, because we will use the fact that $\mathscr{T}_j$ could be rewritten in terms of X_j and $X_j : \mathscr{P} \to \mathbb{R}^m$ and $p_2 - p_1 \in \mathbb{R}^m$. From ***Theorem 1*** in [16], we know that the following should be satisfied:

$$\begin{aligned}
&\left\| \mathscr{T}_j.p_2 - \mathscr{T}_j.p_1 \right\| \\
&\quad = \left\| p_2 - p_1 + \gamma_j\left(p_2, \varepsilon^j\right) - \gamma_j\left(p_1, \varepsilon^j\right) \right\| \\
&\quad \leq \left\| p_2 - p_1 \right\| + \left\| \gamma_j\left(p_2, \varepsilon^j\right) - \gamma_j\left(p_1, \varepsilon^j\right) \right\| \\
&\quad \leq \left\| p_2 - p_1 \right\| + L_j \left\| p_2 - p_1 \right\| \\
&\quad = (1 + L_j) \left\| p_2 - p_1 \right\|,
\end{aligned} \tag{112}$$

where $L_j < 1$ is the Lipschitz constant for $\gamma^j(., \varepsilon^j)$ and $\gamma_j\left(., \varepsilon^j\right)$ is as defined in (32). Let $\left\| X_j(p_2) - X_j(p_1) \right\| \leq K_j \left\| p_2 - p_1 \right\|$, i.e. K_j is the Lipschitz constant for X_j. Then:

$$\left\| (X_j)^n|_{p_2} - (X_j)^n|_{p_1} \right\| \leq K_j \left\| (X_j)^{n-1}|_{p_2} - (X_j)^{n-1}|_{p_1} \right\| \cdots \leq K_j^n \left\| p_2 - p_1 \right\|. \tag{113}$$

By using the equality in (32) and the outcome from (113), (112) can be reformulated as follows:

$$\begin{aligned}
\left\| \mathscr{T}_j.p_2 - \mathscr{T}_j.p_1 \right\| &\leq \left\| p_2 - p_1 \right\| + \left\| \gamma_j\left(p_2, \varepsilon^j\right) - \gamma_j\left(p_1, \varepsilon^j\right) \right\| \\
&= \left\| p_2 - p_1 \right\| + \left\| \varepsilon^j \left(X_j|_{p_2} - X_j|_{p_1} \right) + \frac{\left(\varepsilon^j\right)^2}{2!} \left((X_j)^2|_{p_2} - (X_j)^2|_{p_1} \right) + \ldots \right\| \\
&\leq \left\| p_2 - p_1 \right\| + \left\| \varepsilon^j \left(X_j|_{p_2} - X_j|_{p_1} \right) \right\| + \left\| \frac{\left(\varepsilon^j\right)^2}{2!} \left((X_j)^2|_{p_2} - (X_j)^2|_{p_1} \right) \right\| + \ldots \\
&\leq \left\| p_2 - p_1 \right\| + \varepsilon^j K_j \left\| p_2 - p_1 \right\| + \frac{\left(\varepsilon^j\right)^2}{2!} K_j^2 \left\| p_2 - p_1 \right\| + \ldots \\
&= e^{\left(\varepsilon^j K_j\right)} \left\| p_2 - p_1 \right\|.
\end{aligned} \tag{114}$$

Comparing (112) against (114), we can conclude that:

$$1 + L_j = e^{\left(\varepsilon^j K_j\right)} \implies L_j = e^{\left(\varepsilon^j K_j\right)} - 1. \tag{115}$$

However, since $L_j < 1$:

$$e^{\left(\varepsilon^j K_j\right)} - 1 < 1 \implies \varepsilon^j K_j < ln(2) \implies K_j < \frac{ln(2)}{\varepsilon^j}. \tag{116}$$

F The Order of Magnitude assoc. with the Error in Proposition 3

Our goal is to show that the following ((48) in Subsection 4.3) is equivalent to the statement in (51) in Proposition 3:

$$\Delta \bar{u}_{\bar{x}^{i_1} \ldots \bar{x}^{i_{k-1}}} = \sum_{l=1}^{d} \Delta \bar{x}^l \left[\mathscr{T}_j.p\right]^{u_{x^{i_1} \ldots x^{i_{k-1}} x^l}} + \Theta\left(\left\| \boldsymbol{\Delta} \bar{\boldsymbol{\mathcal{X}}} \left(\boldsymbol{\Delta}^q \boldsymbol{\mathcal{X}}, \varepsilon^j \right) \right\|^2 \right), \forall p \in \mathscr{N}, \tag{117}$$

where $j = 1, \ldots, r$, $k = 1, \ldots, O$, $\boldsymbol{\Delta} \bar{\boldsymbol{\mathcal{X}}} = \left[\Delta \bar{x}^1, \ldots, \Delta \bar{x}^d\right]^T$, and $\boldsymbol{\Delta}^q \boldsymbol{\mathcal{X}} = [0, \ldots, \Delta x^q, 0, \ldots, 0]^T$. Assume that vector field X_j's component functions for independent and dependent directions are as given in (50). Assume further that $|\Delta x^q|$ and $\left|\varepsilon^j\right|$ are sufficiently small, where $q = 1, \ldots, d$, $j = 1, \ldots, r$, d is the dimension of the domain in which solutions reside, and $r = \dim(\mathfrak{g})$. By exploiting the fact in (49):

$$\begin{aligned} \bar{u}_{\bar{x}^{i_1}\ldots\bar{x}^{i_{k-1}}}\left(p+\Delta^q p, \varepsilon^j\right) &= u_{x^{i_1}\ldots x^{i_{k-1}}}\left(\boldsymbol{\mathcal{X}}+\boldsymbol{\Delta}^q\boldsymbol{\mathcal{X}}\right)+\varepsilon^j\eta^{x^{i_1}\ldots x^{i_{k-1}}}\left(p+\Delta^q p\right)+\Theta\left(\left(\varepsilon^j\right)^2\right) \\ \bar{u}_{\bar{x}^{i_1}\ldots\bar{x}^{i_{k-1}}}\left(p, \varepsilon^j\right) &= u_{x^{i_1}\ldots x^{i_{k-1}}}\left(\boldsymbol{\mathcal{X}}\right)+\varepsilon^j\eta^{x^{i_1}\ldots x^{i_{k-1}}}\left(p\right)+\Theta\left(\left(\varepsilon^j\right)^2\right), \end{aligned} \tag{118}$$

where $\boldsymbol{\mathcal{X}} = \left[x^1, \ldots, x^d\right]^T$, and $\Delta^q p$ is as defined in (52). Moreover, since we suppose that u is sufficiently smooth:

$$u_{x^{i_1}\ldots x^{i_{k-1}}}\left(\boldsymbol{\mathcal{X}}+\boldsymbol{\Delta}^q\boldsymbol{\mathcal{X}}\right)-u_{x^{i_1}\ldots x^{i_{k-1}}}\left(\boldsymbol{\mathcal{X}}\right)=\boldsymbol{\Delta}^q\boldsymbol{\mathcal{X}}u_{x^{i_1}\ldots x^{i_{k-1}}x^q}\left(\boldsymbol{\mathcal{X}}\right)+\Theta\left(|\Delta x^q|^2\right). \tag{119}$$

We should make good use of two equations in (118) to approximate $\Delta\bar{u}_{\bar{x}^{i_1}\ldots\bar{x}^{i_{k-1}}}$. It is simply subtracting the second equation from the first one in (118). Then substituting RHS of (119) into this result and collecting common terms yield LHS of (117) (*under the assumption* that our group action for $\varepsilon^j > 0$ maps nearby points $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{\mathcal{X}}+\boldsymbol{\Delta}^q\boldsymbol{\mathcal{X}}$ to again nearby points $\bar{\boldsymbol{\mathcal{X}}}$ and $\bar{\boldsymbol{\mathcal{X}}}+\boldsymbol{\Delta}\bar{\boldsymbol{\mathcal{X}}}$ respectively):

$$\Delta\bar{u}_{\bar{x}^{i_1}\ldots\bar{x}^{i_{k-1}}}=\varepsilon^j\left(\eta^{x^{i_1}\ldots x^{i_{k-1}}}\left(p+\Delta^q p\right)-\eta^{x^{i_1}\ldots x^{i_{k-1}}}\left(p\right)\right)+\boldsymbol{\Delta}^q\boldsymbol{\mathcal{X}}u_{x^{i_1}\ldots x^{i_{k-1}}x^q}\left(\boldsymbol{\mathcal{X}}\right)+\Theta\left(|\Delta x^q|^2\right)+\Theta\left(\left(\varepsilon^j\right)^2\right). \tag{120}$$

As for RHS of (117), observe that:

$$\begin{aligned} \bar{x}^l\left(p+\Delta^q p, \varepsilon^j\right) &= x^l+\delta^l_q\Delta x^q+\varepsilon^j\xi^{x^l}\left(p+\Delta^q p\right)+\Theta\left(\left(\varepsilon^j\right)^2\right), \\ \bar{x}^l\left(p, \varepsilon^j\right) &= x^l+\varepsilon^j\xi^{x^l}\left(p\right)+\Theta\left(\left(\varepsilon^j\right)^2\right), \text{ and} \\ \left[\mathscr{T}_j\cdot p\right]^{u_{x^{i_1}\ldots x^{i_{k-1}}x^l}} &= u_{x^{i_1}\ldots x^{i_{k-1}}x^l}\left(\boldsymbol{\mathcal{X}}\right)+\varepsilon^j\eta^{x^{i_1}\ldots x^{i_{k-1}}x^l}\left(p\right)+\Theta\left(\left(\varepsilon^j\right)^2\right), \end{aligned} \tag{121}$$

by again using (49), where δ^l_q is Dirac delta function. Like in (120), to obtain $\Delta\bar{x}^l$ we subtract the second equation from the first one in (121):

$$\Delta\bar{x}^l=\delta^l_q\Delta x^q+\varepsilon^j\left(\xi^{x^l}\left(p+\Delta^q p\right)-\xi^{x^l}\left(p\right)\right)+\Theta\left(\left(\varepsilon^j\right)^2\right). \tag{122}$$

Therefore, RHS of (117) becomes:

$$\begin{aligned} &\sum_{l=1}^{d}\underbrace{\left(\delta^l_q\Delta x^q+\varepsilon^j\left(\xi^{x^l}\left(p+\Delta^q p\right)-\xi^{x^l}\left(p\right)\right)\right)}_{\approx\Delta\bar{x}^l}\underbrace{\left(u_{x^{i_1}\ldots x^{i_{k-1}}x^l}\left(\boldsymbol{\mathcal{X}}\right)+\varepsilon^j\eta^{x^{i_1}\ldots x^{i_{k-1}}x^l}\left(p\right)\right)}_{\approx\left[\mathscr{T}_j\cdot p\right]^{u_{x^{i_1}\ldots x^{i_{k-1}}x^l}}}+ \\ &\quad\Theta\left(\left(\varepsilon^j\right)^2\right)+\Theta\left(\|\boldsymbol{\Delta}\bar{\boldsymbol{\mathcal{X}}}\left(\boldsymbol{\Delta}^q\boldsymbol{\mathcal{X}},\varepsilon^j\right)\|^2\right). \end{aligned} \tag{123}$$

Collecting $\Theta\left(\left(\varepsilon^j\right)^2\right)$ introduced in both (120) and (123) on LHS, and doing necessary cancellations in order, we obtain:

$$\begin{aligned} &\varepsilon^j\left(\eta^{x^{i_1}\ldots x^{i_{k-1}}}\left(p+\Delta^q p\right)-\eta^{x^{i_1}\ldots x^{i_{k-1}}}\left(p\right)\right)+\Theta\left(|\Delta x^q|^2\right)+\Theta\left(\left(\varepsilon^j\right)^2\right)= \\ &\quad\varepsilon^j\left\{\sum_{l=1}^{d}u_{x^{i_1}\ldots x^{i_{k-1}}x^l}\left[\xi^{x^l}\left(p+\Delta^q p\right)-\xi^{x^l}\left(p\right)\right]+\Delta x^q\eta^{x^{i_1}\ldots x^{i_{k-1}}x^q}\left(p\right)\right\}+\Theta\left(\|\boldsymbol{\Delta}\bar{\boldsymbol{\mathcal{X}}}\left(\boldsymbol{\Delta}^q\boldsymbol{\mathcal{X}},\varepsilon^j\right)\|^2\right), \end{aligned} \tag{124}$$

where we introduce additional errors $\Theta\left(|\Delta x^q|^2\right)$ and $\Theta\left(\left(\varepsilon^j\right)^2\right)$ due to approximation in (118), (119), and (121). It could be realized that as Δx^q, $\varepsilon^j \to 0$, equality in (124) is satisfied. On the other hand; only if $\varepsilon^j \to 0$, then the group action disappears and we are only left with (see (122) for what happens to $\Theta\left(\|\boldsymbol{\Delta}\bar{\boldsymbol{\mathcal{X}}}\left(\boldsymbol{\Delta}^q\boldsymbol{\mathcal{X}},\varepsilon^j\right)\|^2\right)$ when $\varepsilon^j \to 0$):

$$\Theta\left(|\Delta x^q|^2\right)=\Theta\left(|\Delta x^q|^2\right). \tag{125}$$

Furthermore; only if $\Delta x^q \to 0$, then:

$$\Theta\left(\left(\varepsilon^j\right)^2\right)\neq 0, \tag{126}$$

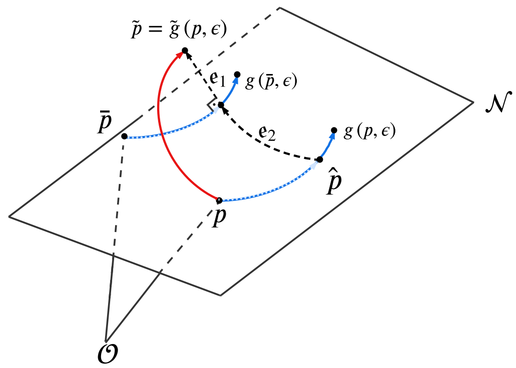

Figure 20. Numerical errors induced by translating a solution u to $\tilde{u}$ once the translation $\tilde{p} = \tilde{g}(p, \varepsilon)$ is executed in prolonged space $\mathscr{P}$. Two types of error occur: (i) $\mathfrak{e}_1$: Diverging from solution manifold $\mathscr{N}$, which violates our constraint to stay on the manifold, and (ii) $\mathfrak{e}_2$: Moving away from the correct integral curve generated by true group action $g(p, \varepsilon)$, which violates the translated solution's consistency with its derivatives.

is obtained. Frankly, this is the expected behaviour of this approximation scheme. Notice that the approximation errors $\Theta\left(\left(\varepsilon^j\right)^2\right)$ induced in (120) and (122) die out as $\Delta x^q \to 0$ regardless of the value of ε^j. However, the last equation in (121) has no $\boldsymbol{\Delta}^q \boldsymbol{\mathcal{X}}$ term involved and hence, its error term $\Theta\left(\left(\varepsilon^j\right)^2\right)$ is not affected by any behaviour of Δx^q.

In summary, we assume that the error terms $\Theta\left(|\Delta x^q|^2\right)$, $\Theta\left(\left(\varepsilon^j\right)^2\right)$, and $\Theta\left(\|\boldsymbol{\Delta}\bar{\boldsymbol{\mathcal{X}}}\left(\boldsymbol{\Delta}^q \boldsymbol{\mathcal{X}}, \varepsilon^j\right)\|^2\right)$ are *negligible* **if** $|\Delta x^q|$ and $\left|\varepsilon^j\right|$ are sufficiently small. Thus, ignoring these error terms, we try to approximate LHS of (124) to its RHS up to a certain threshold.

G The Errors Induced by Group Actions & The Retraction Map

Due to various truncation errors, numeric errors inherent to sample dataset generation, etc.; our one-parameter group action $\tilde{g}(p, \varepsilon)$ (i.e. $\mathscr{T}.p$) is not ideal. We can summarize all these errors in two broad categories: (i) Violating our constraint $\mathscr{C}_{1_j}$ in (55) and (ii) Violating our consistency cost $\mathscr{CC}$ specified in (53). Both errors are depicted in Figure-20 as $\mathfrak{e}_1$ and $\mathfrak{e}_2$ in order. Our experiments, as discussed in 5.4, show us that the consistency of translated solution is not affected as much as constraints that enforce solutions to stay on the manifold $\mathscr{N}$, thanks to $\mathscr{CC}$ introduced in (58). In fact, it could be ignored once projection onto $\mathscr{N}$ is properly done. Therefore, in this study we do not focus on fixing $\mathfrak{e}_2$ after every translation; but rather on fixing $\mathfrak{e}_1$, which spirals out of control upon thousands of translations as shown in Table-2 and Table-3.

This could be mitigated by a process called **"retraction"** in differential geometry. It is an idea based on "*Whitney approximation theorem*" for functions. By **Proposition 6.25** in [33], any embbeded manifold has a retraction map from its tubular neighbourhood (see Figure-10) to the manifold itself, i.e. $r: U \to \mathscr{N}$, where $U \subset \boldsymbol{\mathcal{R}}^m$ is an ε-neighbourhood lying within tubular neighbourhood of $\mathscr{N}$. Besides, the manifolds we are interested in are embedded into $\mathscr{P}$. In the case of linear DEs, r is as simple as a projection from U to $\mathscr{N}$, i.e.:

$$r\left(s_k^i\right) = \bar{s}_k^i = s_k^i - \langle s_k^i, \vec{n}\rangle\, \vec{n}, \tag{127}$$

where $\langle , \rangle$ is inner product and $\vec{n}$ is the unit normal vector for hyperplane generated by a given set of linear DEs and $i = 1, \ldots, N$, and N stands for how many times group action is applied in a subsequent manner. For instance, in the case of wave equation in 2-D $\vec{n}|_p = [0\,0 \ldots 0\,1\,0\,0\,1\,0 \,{}^{-1}\!/_{c^2}]$, where c is DE constant and $p \in \mathscr{N}$. The same experiments in 5.4 are conducted with retraction map, r, included this time. At every i^{th} step, the output of group action, g_j, is subject to retraction as given below:

$$s_k^i = g_j . s_k^{i-1} \text{ then, } \bar{s}_k^i = r\left(s_k^i\right). \tag{128}$$

Results are summarized in Table-4 and Table-5 respectively. It is obvious that retraction strives to keep points on the manifold, $\mathscr{N}$, and hence output of subsequent group actions does not diverge from $\mathscr{N}$.

	Subgroup Idx (j)	$\mathcal{C}_{1_j}$	$\mathcal{CC}$
#g=1			
	2	3.391e-13	8.817e-11
	3	1.605e-13	5.143e-11
	4	1.994e-13	5.205e-12
	5	3.317e-13	1.811e-12
	6	3.081e-13	4.720e-12
	7	1.846e-13	7.727e-12
	8	2.064e-13	1.700e-15
	9	1.379e-13	3.750e-12
	10	1.757e-13	9.372e-11
	11	1.341e-13	1.831e-14
	12	1.923e-13	2.657e-13
#g=1000			
	2	2.619e-13	8.817e-11
	3	3.369e-13	5.143e-11
	4	2.135e-13	5.205e-12
	5	2.590e-13	1.811e-12
	6	2.119e-13	4.720e-12
	7	2.572e-13	7.727e-12
	8	1.664e-13	1.700e-15
	9	1.339e-13	3.750e-12
	10	3.230e-13	9.372e-11
	11	1.305e-13	1.831e-14
	12	3.439e-13	2.657e-13

Table 4. Quality of generalization in the case of spatial region shift with retraction, where #g stands for number of group actions applied to the initial solution. Compare this against the entries in Table-2.

	Subgroup Idx (j)	$\mathcal{C}_{1_j}$	$\mathcal{CC}$
#g=1			
	2	2.950e-13	8.116e-10
	3	3.212e-13	5.136e-10
	4	3.789e-13	2.833e-11
	5	1.577e-13	3.984e-12
	6	1.984e-13	3.705e-11
	7	1.508e-13	4.608e-11
	8	2.714e-13	6.238e-14
	9	2.369e-13	2.662e-11
	10	2.571e-13	5.485e-10
	11	2.089e-13	1.049e-12
	12	2.060e-13	4.619e-12
#g=1000			
	2	2.738e-13	8.116e-10
	3	2.808e-13	5.136e-10
	4	1.126e-13	2.833e-11
	5	1.537e-13	3.984e-12
	6	3.508e-13	3.705e-11
	7	1.439e-13	4.608e-11
	8	2.762e-13	6.238e-14
	9	2.231e-13	2.662e-11
	10	3.249e-13	5.485e-10
	11	3.208e-13	1.049e-12
	12	3.915e-13	4.619e-12

Table 5. Quality of generalization in the case of initial condition u_{t_0} set to (87) with retraction, where #g stands for number of group actions applied to the initial solution. Compare this against the entries in Table-3.

H Algorithms

Algorithm 1: Computing $(X_j|_p)^n$ for $n \geq 2$

```
def computeVecFiAct(X_j|_p, actGrad):
    // Take inner prod. to calc. the action of X_j|_p on itself for
    // k times, where k is the order of derivatives stored in actGrad
    for i in range(1..n):
        actVF[i] ← dotProd(X_j|_p, actGrad[i]);
    return actVF
def computeAllOrders(expOrder, p, j):
    global actionsList ← [];
    // The first order is equal to the evaluation of X_j|_p itself
    X_j|_p ← iResNETBlock(j, p);
    actionList.append(X_j|_p);
    for o in range(1..expOrder):
        gradP[i] ← grad(actionList[i − 1], p);
        (X_j|_p)^o ← computeVecFiAct(X_j|_p, gradP);
        actionsList.append((X_j|_p)^o);
    return actionList
```

Algorithm 2: The cost function to be minimized for enforcing linear independence of $\{X_1, \ldots, X_r\}|_p$

```
def costFuncAtPt(p):
    // Extract X_1|_p and X_2|_p
    X_1|_p ← iResNETBlock(1, p);
    X_2|_p ← iResNETBlock(2, p);
    // To span 𝒱_2
    𝒥|_p ← ( ⟨X_1|_p, X_2|_p⟩ / (‖X_1|_p‖_2 ‖X_2|_p‖_2) )^2;
    for i in range(3..n):
        X_i|_p ← iResNETBlock(i, p);
        X_∥i|_p ← M_{i−1}|_p X_i|_p;
        // To span 𝒱_i in an incremental fashion
        𝒥|_p ← 𝒥|_p + ( ⟨X_i|_p, X_∥i|_p⟩ / (‖X_i|_p‖_2 ‖X_∥i|_p‖_2) )^2;
    return 𝒥|_p
```

Algorithm 3: Training loop for learning symmetries

```
def nrmCost(L, R):
    N ← (L−R)^2 / ((|L|+|R|)^2 + δ);
    return N;
def measureLI(s_k, G_θ, j):
    X_j|_p ← iResNETBlock(j, s_k);
    X_∥j|_p ← M_{j−1} X_j|_p;
    linIndCost ← ( ⟨X_j|_p, X_∥j|_p⟩ / (‖X_j|_p‖_2 ‖X_∥j|_p‖_2) )^2;
    return linIndCost;
def cmpMinMax(value, minRef, maxRef):
    if maxRef.value < 0.0 or maxRef.value < value:
        maxRef.value ← value;
    elif minRef.value < 0.0 or minRef.value > value:
        minRef.value ← value;
def trainSubgrp(Γ, G_θ, j, Γ̃):
    bestModel ← None;
    for b_q ← nextBatch(Γ):
        max_{C1_j}, min_{C1_j} max_{C2_j}, min_{C2_j} ← −1.0;
        finalCost_j ← zeros(|b_q|);
        for s_k ← nextItem(b_q):
            // no other subgroup to measure linear-ind
            if j > 0:
                J_j|_{s_k} ← measureLI(s_k, G_θ, j);
            else:
                J_j|_{s_k} ← 0.0;
            C1_j|_{s_k} ← max {‖D(T_j.s_k)‖_2 − κ_1, 0.0};
            cmpMinMax(C1_j|_{s_k}, max_{C1_j}, min_{C1_j});
            C2_j|_{s_k} ← κ_2 − [σ_{1j}, ..., σ_{ij}, ..., σ_{N_j j}]^T|_{s_k};
            cmpMinMax(C2_j|_{s_k}, max_{C2_j}, min_{C2_j});
            RI_j^LHS|_{s_k} ← 1/(m.(j−1)) Σ_{i=1}^{j−1} Σ_{q=1}^{m} RI_ij^{LHS_q}|_{s_k};
            RI_j^RHS|_{s_k} ← 1/(m.(j−1)) Σ_{i=1}^{j−1} Σ_{q=1}^{m} RI_ij^{RHS_q}|_{s_k};
            R̃I_j|_{s_k} ← nrmCost(RI_j^LHS|_{s_k}, RI_j^RHS|_{s_k});
            CC_j^LHS|_{s_k} ← 1/d Σ_{q=1}^{d} CC_j^{LHS_q}|_{s_k};
            CC_j^RHS|_{s_k} ← 1/d Σ_{q=1}^{d} CC_j^{RHS_q}|_{s_k};
            C̃C_j|_{s_k} ← nrmCost(CC_j^LHS|_{s_k}, CC_j^RHS|_{s_k});
            J̃_j^I|_{s̄_k} ← nrmCost(T_j^{−1}.s̄_k, s_k);
            ex_{C1_j} ← (max_{C1_j} + min_{C1_j}) / 2.0;
            ex_{C1_j} ← (max_{C2_j} + min_{C2_j}) / 2.0;
            for k in range(|b_q|):
                C̃1_j|_{s_k} ← σ̄(ex_{C1_j}) ϒ(C1_j|_{s_k});
                C̃2_j|_{s_k} ← Σ_{i=1}^{N_j} σ̄(ex_{C2_j}) ϒ(C2_ij|_{s_k});
                finalCost_j[k] ← J_j|_{s_k} + R̃I_j|_{s_k} + J̃_j^I|_{s̄_k} + C̃C_j|_{s_k} + ψ̃_t(C̃1_j|_{s_k}) + ψ̃_t(C̃2_j|_{s_k});
            finalCost_j ← mean(finalCost_j);
            finalCost_j.backward();
        // end of inner for-loop
        bestModel ← validate(G_θ, Γ̃);
    return bestModel;
```

Algorithm 3: Training loop for learning symmetries (cnt.)

```
def mainLoop(Γ, Γ̃, G_θ, TRIES):
    numTries ← TRIES;
    failed ← [];
    grpsToTrain ← G_θ;
    while len (grpsToTrain) > 0 and numTries > 0:
        for G_j in grpsToTrain:
            setRandomSeed(j + numTries);
            bestModel ← None;
            for ep in range(numEpoch):
                bestModel ← trainSubgrp(Γ, G_θ, j, Γ̃);
            // Freeze its weights and finalize its training
            if not bestModel:
                failed.append(j);
            else:
                finalize(G_θ[j]);
            // Reinitialize the entire G_θ; but substitute weights
            // of subgroups which succeed to be trained
            G_θ ← reinitialize(j, failed);
        grpsToTrain ← failed;
        numTries ← numTries − 1;
        failed ← [];
    return G_θ
```