A Proof

Proof of Theorem 2. To see $H^o$ is still an RKHS under the new norm, note that a Hilbert space is an RKHS if point evaluation functionals are bounded, i.e., there exists a $C > 0$ such that for any $x \in X$ and $f$ in the space, it holds that $|f(x)| \leq C \|f\|$. Since $H$ is an RKHS, so $|f(x)| \leq C \|f\|_H$. Since $\|f\|_H \geq \|f\|_{H^o}$, it follows trivially that $|f(x)| \leq C \|f\|_{H^o}$. So $H^o$ is an RKHS.

Clearly $k^o(x, \cdot) = k(x, \cdot) - z(x)^T (I + K_z)^{-1} z(\cdot)$ is in $H^o$ as it linearly combines $k(x, \cdot)$ and $\{z_i\}$ which are in $H$, and $H^o$ consists of the same set of functions as $H$. So it suffices to show $k^o(x, \cdot)$ is a representer of point evaluation at $x$ in $H^o$.

For any $f \in H^o$ (or equivalently $f \in H$), denote $z_f := (\langle z_1, f \rangle_H, \ldots, \langle z_m, f \rangle_H)^T$. It follows that for all $x \in X$,

$$
\langle k^o(x, \cdot), f \rangle_{H^o} = \langle k^o(x, \cdot), f \rangle_H + \sum \langle z_i, f \rangle_H \langle z_i, k^o(x, \cdot) \rangle_H \quad \text{(by the definition of $\langle \cdot, \cdot \rangle_{H^o}$)}
$$

$$
= \langle k(x, \cdot) - z(x)^T (I + K_z)^{-1} z(\cdot), f \rangle_H + \sum \langle z_i, f \rangle_H \langle z_i, k(x, \cdot) - z(x)^T (I + K_z)^{-1} z(\cdot) \rangle_H
$$

$$
= f(x) - z(x)^T (I + K_z)^{-1} z_f + (z(x)^T z_f - z(x)^T (I + K_z)^{-1} K_z z_f) f(x),
$$

where the last equality follows from the simple fact that $-(I + K_z)^{-1} + I - (I + K_z)^{-1} K_z = 0$ (to see it, left multiply both sides by the invertible matrix $I + K_z$). So $k^o$ is the reproducing kernel of $H^o$.

Proof of Theorem 3. Let $v := \sum_i \alpha_i z(x_i) - \sum_j \beta_j z(y_j)$. Then

$$
0 \leq \left\| \sum_i \alpha_i \varphi^o(x_i) - \sum_j \beta_j \varphi^o(y_j) \right\|_{H^o}^2
$$

$$
= \sum_i \alpha_i \alpha_i k^o(x_i, x_i') + \sum_{j,j'} \beta_j \beta_j' k^o(y_j, y_{j'}) - 2 \sum_i \alpha_i \beta_j k^o(x_i, y_j)
$$

by (3)

$$
= \sum_i \alpha_i \alpha_i (k(x_i, x_i') - z(x_i)^T M z(x_i')) \quad \text{where} \quad M := (I + K_z)^{-1}
$$

$$
+ \sum_{j,j'} \beta_j \beta_j' (k(y_j, y_{j'}) - z(y_j)^T M z(y_{j'})) - 2 \sum_i \alpha_i \beta_j (k(x_i, y_j) - z(x_i)^T M z(y_j))
$$

$$
= \left\| \sum_i \alpha_i \varphi(x_i) - \sum_j \beta_j \varphi(y_j) \right\|_H^2 - v^T M v = -v^T M v \leq 0,
$$

So we conclude $\left\| \sum_i \alpha_i \varphi^o(x_i) - \sum_j \beta_j \varphi^o(y_j) \right\|_{H^o} = 0$, i.e., $\sum_i \alpha_i \varphi^o(x_i) = \sum_j \beta_j \varphi^o(y_j)$.

Property 1. The warping operator is non-expansive.

Proof. For any $f = \sum_i \alpha_i \varphi(x_i)$, denote $z(x_i) = (z_1(x_i), \ldots, z_m(x_i))^T$. Then

$$
\|f^o\|^2_{H^o} = \left\| \sum_i \alpha_i \varphi^o(x_i) \right\|_{H^o}^2 = \left\| \sum_{ij} \alpha_i \alpha_j k^o(x_i, x_j) \right\|_{H^o}^2
$$

$$
= \sum_{ij} \alpha_i \alpha_j (k(x_i, x_j) - z(x_i)^T (I + K_z)^{-1} z(x_j))
$$

$$
= \|f\|^2_H - \left( \sum_i \alpha_i z(x_i) \right)^T (I + K_z)^{-1} \left( \sum_j \alpha_j z(x_j) \right) \leq \|f\|^2_H.
$$

So $\|f^o\|_{H^o} \leq \|f\|_H$ as $K_z$ is PSD.
Proof of Lemma 1. The proof follows that of Proposition 4 in [29], but inserts $\|W_{k-1}, L_r\|$ as needed. Define $(MPAW)_{k:j} := M_k P_k A_k 1 W_{k-1} M_{k-1} P_{k-1} A_{k-2} W_{k-2} \ldots M_j P_j A_j 1 W_j$. Noting that $\|A_k\| \leq 1$, $\|P_k\| = 1$, $W_k$ and $M_k$ are non-expansive, we obtain
\begin{align*}
&\|\Psi_n(L_r x) - \Psi_n(x)\| \leq \|A_n(MPAW)_{n:2} M_1 P_1 A_0 L_r x - A_n(MPAW)_{n:2} M_1 P_1 A_0 x\| \\
&\leq \|A_n(MPAW)_{n:2} M_1 P_1 A_0 L_r x - A_n(MPAW)_{n:2} M_1 L_r P_1 A_0 x\| \\
&\quad + \|A_n(MPAW)_{n:2} M_1 L_r P_1 A_0 x - A_n(MPAW)_{n:2} M_1 P_1 A_0 x\|
\end{align*}
(a)\hspace{1cm} (27)
(b)\hspace{1cm} (28)
(c)\hspace{1cm} (29)
\begin{align*}
&\leq \|P_1 A_0, L_r||x|| + \|A_n(MPAW)_{n:2} L_r M_1 P_1 A_0 x - A_n(MPAW)_{n:2} M_1 P_1 A_0 x\|
\end{align*}
(30)
\begin{align*}
&\leq \|P_1 A_0, L_r||x|| + \|A_n(MPAW)_{n:3} M_2 P_2 A_1 W_1 L_r y_1 - A_n(MPAW)_{n:3} M_2 P_2 A_1 L_r W_1 y_1\|
\quad + \|A_n(MPAW)_{n:3} M_2 P_2 A_1 L_r W_1 y_1 - A_n(MPAW)_{n:3} M_2 P_2 A_1 y_1\|
\end{align*}
(31)
(32)
(33)
\begin{align*}
&\leq \|P_1 A_0, L_r||x|| + \|W_1, L_r||x|| + \|A_n(MPAW)_{n:3} M_2 P_2 A_1 L_r z_1 - A_n(MPAW)_{n:3} M_2 P_2 A_1 z_1\|
\end{align*}
(34)
Here (a) is by $M_1 L_r = L_r M_1$, (b) is by defining $y_1 = M_1 P_1 A_0 x$, (c) is by defining $z_1 = W_1 y_1$. Noting that the last line is isomorphic to the first line and $\|z_1\| \leq \|x\|$, we can unfold this recursion and prove Lemma 1. □

Proof of Theorem 4. We use Schur’s test by reformulating $[W, L_r]$ as an integral operator and bounding its kernel [Lemma A.1, 29]. Letting $\xi = (I - \tau)^{-1}$ and noting the Jacobian in change of variable for integral, we have
\begin{align}
[W, L_r] f(z) = W L_r f(z) - L_r W f(z) \\
= \int W(z, u) f(u - \tau(u)) du - \int W(z - \tau(z), u) f(u) du \\
= \int W(z, \xi(s)) f(s) \frac{du}{ds} ds - \int W(z - \tau(z), s) f(s) ds
\end{align}
(35)
(36)
(37)
Noting that $\alpha := \frac{\|\tau\|}{\|\xi\|} = \text{det}(I - \nabla \tau(u))^{-1}$, we derive the kernel
\begin{align}
k(z, s) = & \alpha W(z, \xi(s)) - W(z - \tau(z), s) \\
= & (\alpha - 1) W(z, \xi(s)) + W(z, \xi(s)) - W(z, s) + W(z, s) - W(z - \tau(z), s).
\end{align}
(38)
(39)
Since $\text{det}(I - \nabla \tau(u)) \geq (1 - \|\nabla \tau\|_\infty)^d \geq 1 - d \|\nabla \tau\|_\infty$, it follows that $\alpha \in [1, 1 + 2d \|\nabla \tau\|_\infty]$. We can then bound each term as
\begin{align}
|A| \leq 2d \|\nabla \tau\|_\infty \|W(z, \xi(s))\| \leq 2d \|\nabla \tau\|_\infty, \\
|B| \leq L_w \|\xi(s) - s\| = L_w \|\tau(\xi(s))\| \leq L_w \|\tau\|_\infty, \\
|C| \leq L_w \|\tau\|_\infty,
\end{align}
(40)
(41)
(42)
where (a) is because $W$ is non-expansive. As $\Omega$ is bounded, we can bound both $\int |k(z, s)| dz$ and $\int |k(z, s)| ds$ by $C_1 \|\nabla \tau\|_\infty + C_2 L_w \|\tau\|_\infty$, where $C_1$ and $C_2$ depend on $\Omega$ only. Then Schur’s test directly implies (18). □

B Derivative for End-to-end Training of Single Hidden Layer Network

Suppose we have invariance representers with finite approximation $Z = \{z_1, \ldots, z_m\}$ (we dropped the tilde to simplify notation as here we only deal with finite approximations). Let there be $n_c$ classes, and the output layer weight be a matrix $O$ with each column corresponding to a class. Let $\xi(x)$ be the FA of $x$ using the Fourier samples $B := (\omega_1, \ldots, \omega_p)$. Then the end-to-end empirical risk minimization can be written as
\begin{align}
\min_{B,O} \mathbb{E}_{(x,l) \sim \tilde{p}} \left[L(O^\top (I + ZZ^\top)^{-1/2} \xi(x), l)\right],
\end{align}
(43)
where $\tilde{p}$ is the empirical distribution over feature/label pair $(x, l)$. Both $\xi(x)$ and $Z$ depend on $B$.
Denote \( f(B, O) = L(O^\top (I + ZZ^\top))^{-1/2}x, l \). Then trivially
\[
\nabla_O f(B, O) = (I + ZZ^\top)^{-1/2}x \cdot r^\top,
\]
where \( r := \nabla L(O^\top (I + ZZ^\top)^{-1/2}x, l) \in \mathbb{R}^{n_c} \) and \( \nabla L \) denotes the partial derivative of \( L \) with respect to its first argument.

To compute the derivative in \( B \), we analyze the change of \( f \) when \( B \) is perturbed by \( \Delta B \) with \( \|\Delta B\| := \sum_i |\omega_i| \leq \epsilon \). Suppose \( \Delta Z \) is the corresponding change of \( Z \) up to \( o(\epsilon) \). Then letting \( M^2 := I + ZZ^\top \) and \( G := (\Delta Z)Z^\top + Z(\Delta Z)^\top \), we have
\[
(I + (Z + \Delta Z)(Z + \Delta Z)^\top)^{-1/2} = (I + ZZ^\top + G + o(\epsilon))^{-1/2}
\]
\[
= [M(I + M^{-1}GM^{-1} + o(\epsilon))M]^{-1/2}
\]
\[
= M^{-1/2}(I + M^{-1}GM^{-1} + o(\epsilon))M^{-1/2}
\]
\[
= M^{-1/2}(I - \frac{1}{2}M^{-1}GM^{-1} + o(\epsilon))M^{-1/2}
\]
\[
= M^{-1} - \frac{1}{2}M^{-2}GM^{-2} + o(\epsilon).
\]

The change of \( \xi(x) \) with respect to \( \Delta B \) depends on the kernel. Let us use Gaussian kernel and
\[
\xi_B(x) = \frac{1}{\sqrt{p}} \begin{pmatrix}
\cos(\omega_1^\top x) \\
\sin(\omega_1^\top x) \\
\vdots \\
\cos(\omega_p^\top x) \\
\sin(\omega_p^\top x)
\end{pmatrix}
\Rightarrow \Delta \xi_B(x) = \frac{1}{\sqrt{p}} \begin{pmatrix}
-\sin(\omega_1^\top x) \cdot x^\top \Delta \omega_1 \\
\cos(\omega_1^\top x) \cdot x^\top \Delta \omega_1 \\
\vdots \\
-\sin(\omega_p^\top x) \cdot x^\top \Delta \omega_p \\
\cos(\omega_p^\top x) \cdot x^\top \Delta \omega_p
\end{pmatrix} = H \cdot (\Delta B)^\top \cdot x
\]
where \( H = \frac{1}{\sqrt{p}} \begin{pmatrix}
-\sin(\omega_1^\top x) & 0 & 0 & \ldots & 0 \\
\cos(\omega_1^\top x) & 0 & 0 & \ldots & 0 \\
0 & -\sin(\omega_2^\top x) & 0 & \ldots & 0 \\
0 & \cos(\omega_2^\top x) & 0 & \ldots & 0 \\
\vdots & \vdots & \ddots & \ddots & \vdots
\end{pmatrix} \in \mathbb{R}^{2p \times p}.
\]

Therefore
\[
f(B + \Delta B, g) - f(B, g) = r^\top O^\top \left[ \left( M^{-1} - \frac{1}{2}M^{-2}GM^{-2} \right) \cdot (\xi_B(x) + H \cdot (\Delta B)^\top \cdot x) - M^{-1} \xi_B(x) + o(\epsilon) \right]
\]
\[
= r^\top O^\top \left( M^{-1}H \cdot (\Delta B)^\top \cdot x - \frac{1}{2}M^{-2}GM^{-2}\xi_B(x) + o(\epsilon) \right).
\]

The contribution of the first part to the gradient in \( B \) is easy to derive because
\[
r^\top O^\top M^{-1}H \cdot (\Delta B)^\top \cdot x = \langle \Delta B, xr^\top O^\top M^{-1}H \rangle.
\]

So the contribution to the gradient aggregated over the entire dataset is \( \sum_x x_r^\top O^\top M^{-1}H_x \) (note \( r \) and \( H \) depend on \( x \)). \( M \) (independent of \( x \)) can be computed by first finding the left singular vectors of \( Z \) (or a few leading ones), and then adjusting their corresponding singular values to give the eigen-decomposition of \( M \).

The second term in (53) can be expanded as
\[
-\frac{1}{2}r^\top O^\top M^{-2}GM^{-2}\xi_B(x) = -\frac{1}{2}r^\top O^\top M^{-2}(Z^\top Z^\top + Z(\Delta Z)^\top)M^{-2}\xi_B(x)
\]
\[
= -\frac{1}{2} (a^\top (\Delta Z)b + c^\top (\Delta Z)d) ,
\]
where \( a = M^{-\frac{1}{2}}Or, b = Z^\top M^{-\frac{1}{2}}\xi_B(x), c = M^{-\frac{1}{2}}\xi_B(x), d = Z^\top M^{-\frac{1}{2}}Or. \)
Suppose \( z_i \) models the gradient at \( y_i \) in the direction of \( v_i \). Then by (7),

\[
Z = \frac{1}{\sqrt{P}} \begin{pmatrix}
-(\omega_1^T v_1) \sin(\omega_1^T y_1) & \ldots & -(\omega_1^T v_m) \sin(\omega_1^T y_m) \\
(\omega_1^T v_1) \cos(\omega_1^T y_1) & \ldots & (\omega_1^T v_m) \cos(\omega_1^T y_m) \\
\vdots & \vdots & \vdots \\
-(\omega_p^T v_1) \sin(\omega_p^T y_1) & \ldots & -(\omega_p^T v_m) \sin(\omega_p^T y_m) \\
(\omega_p^T v_1) \cos(\omega_p^T y_1) & \ldots & (\omega_p^T v_m) \cos(\omega_p^T y_m)
\end{pmatrix}
\]

\Rightarrow \Delta Z = \frac{1}{\sqrt{P}} \begin{pmatrix}
\alpha_1^T \Delta \omega_1 & \ldots & \alpha_m^T \Delta \omega_1 \\
\beta_1^T \Delta \omega_1 & \ldots & \beta_m^T \Delta \omega_1 \\
\vdots & \vdots & \vdots \\
\alpha_1^T \Delta \omega_p & \ldots & \alpha_m^T \Delta \omega_p \\
\beta_1^T \Delta \omega_p & \ldots & \beta_m^T \Delta \omega_p
\end{pmatrix}, \text{ where } \begin{pmatrix}
\alpha_{ij} = -v_j \sin(\omega_i^T y_j) - y_j(\omega_i^T v_j) \cos(\omega_i^T y_j) \\
\beta_{ij} = v_j \cos(\omega_i^T y_j) - y_j(\omega_i^T v_j) \sin(\omega_i^T y_j)
\end{pmatrix}.

(58)

Denote \( a = (a_1^+, a_1^-, \ldots, a_p^+, a_p^-)^T \). Then we can collect the terms in \( a^T (\Delta Z) b \) that involve \( \Delta \omega_i \):

\[
\frac{1}{\sqrt{P}} \left\langle \Delta \omega_i, a_1^T \sum_{j=1}^m \alpha_{ij} b_j + a_1^T \sum_{j=1}^m \beta_{ij} b_j \right\rangle = \frac{1}{\sqrt{P}} \left\langle \Delta \omega_i, \sum_{j=1}^m p_{ij} v_j + \sum_{j=1}^m q_{ij} y_j \right\rangle,
\]

where \( p_{ij} = b_j(-a_1^+ \sin(\omega_i^T y_j) + a_1^- \cos(\omega_i^T y_j)) \)

\( q_{ij} = -b_j(a_1^+ (\omega_i^T v_j) \cos(\omega_i^T y_j) + a_1^- (\omega_i^T v_j) \sin(\omega_i^T y_j)) \).

(60)

So the gradient in \( B \) can be compactly written as \(-\frac{1}{\sqrt{P}} (VP^T + YQ^T)\), where \( V = (v_1, \ldots, v_m) \) and \( Y = (y_1, \ldots, y_m) \). Furthermore, incorporating the contribution from \( c^T (\Delta Z) d \), we can augment \( P \) and \( Q \) into:

\[
p_{ij} = -\sin(\omega_i^T y_j)(a_1^T b_j + c_1^T d_j) + \cos(\omega_i^T y_j)(a_1^- b_j + c_1^- d_j)
\]

\( q_{ij} = -(\omega_i^T v_j) \cos(\omega_i^T y_j)(a_1^+ b_j + c_1^+ d_j) - (\omega_i^T v_j) \sin(\omega_i^T y_j)(a_1^- b_j + c_1^- d_j).\)

(63)

(64)

So finally, the gradient in \( B \) can be computed by \(-\frac{1}{\sqrt{P}} (VP^T + YQ^T)\). The procedure is:

1. Compute all \( \omega_i^T y_j \), followed by their sin and cos. Denote the results by matrices \( T \) (for products), \( S \) (for sine), and \( C \) (for cosine), respectively, all sized \( p \times m \). Also compute \( \omega_i^T v_j \) as a matrix \( R \in \mathbb{R}^{p \times m} \). These cost \( O(pmld) \) where \( d \) is the dimensionality of the input \( x \).

2. Compute \( P \) and \( Q \) by

\[
P = -S \circ (a^T b^T + c^T d^T) + C \circ (a^- b^T + c^- d^T)
\]

\( Q = -R \circ [C \circ (a^T b^T + c^T d^T) + S \circ (a^- b^T + c^- d^T)]\),

where \( \circ \) is the Hadamard product. The total cost is \( O(pmld) \).

3. Compute \(-\frac{1}{\sqrt{P}} (VP^T + YQ^T)\), which costs \( O(pml) \).

If we naively perform this repeatedly for each of the \( l \) training examples, the total cost will be \( O(pml(l+d+1)) \). Fortunately this can be reduced to \( O(pml(d+1)) \) because different training examples only differ in \( a, b, c, d \) vectors, while \( T, S, \) or \( C \) are shared. So overall we can replace step 2 by

\[
P = -S \circ F^+ + C \circ F^- \quad \text{and} \quad Q = -R \circ [C \circ F^+ + S \circ F^-],
\]

where \( F^+ = \sum_x a_x^+ b_x^T + \sum_x c_x^+ d_x^T \)

\( F^- = \sum_x a_x^- b_x^T + \sum_x c_x^- d_x^T \).

(65)

(66)

This costs \( O(pml) \). Of course it still costs to compute \( a, b, c, d \) for all \( x \) too. Once we assemble \( a_x^+, a_x^-, b_x, c_x^+, c_x^-, d_x \) into matrices \( A^+, A^-, E \) (unfortunately the symbol \( B \) has been taken), \( C^+, C^-, D \) by columns, we get

\[
F^+ = A^+ E^T + C^+ D^T, \quad F^- = A^- E^T + C^- D^T,
\]

(69)

again as efficient matrix-matrix multiplications.
C Connection with Convolutional Neural Networks

As demonstrated by [29], CKNs contain a set of convolutional neural networks (CNNs) with smooth and homogeneous activations. We now show that such a relationship is retained when kernel warping is introduced, and the new RKHS norm of the overall function allows CNNs to favor invariance-respecting configurations.

Consider a CNN function $f_\sigma$ that is defined recursively through the layers. The input image $z_0 = x_0$ is in $L^2(\Omega, \mathbb{R}^p)$ (i.e., $p_0$ channels). The image $z_k$ at layer $k$ lies in $L^2(\Omega, \mathbb{R}^{p_k})$, constructed from the previous $z_{k-1}$ using convolution and pooling. In particular, it employs $p_k$ filters $\{w^i_k\}_{i=1}^{p_k}$ where each $w^i_k = \{w^j_{k,i}\}_{j=1}^{p_{k-1}} \in L^2(S_k, \mathbb{R}^{p_{k-1}})$. Then the convolution and activation yield $z^i_k(u) = n_k(u)\sigma(\langle w^j_{k,i}, P_k z_{k-1}(u) \rangle / n_k(u))$ for channel $i \in [p_k] := \{1, 2, \ldots, p_k\}$ and $u \in \Omega$, where $\sigma$ is a smooth function, and $n_k(u) = \|P_k z_{k-1}(u)\|$. Finally, the $k$-th layer image is obtained by pooling, with $z_k = A_k z_k$. A linearly fully connected output/prediction layer is applied to the last layer $n$ by $f_\sigma(x_0) = \langle w_{n+1}, z_n \rangle$.

[29] showed that under smoothness conditions of $\sigma$, $f_\sigma$ with any value of filters $\{w^i_k\}$ can be reconstructed by a CKN with carefully engineered functions lying in the intermediate RKHS $\mathcal{H}_k$. Specifically, the first layer adopts $f^1_1 \in \mathcal{H}_1$ and $g^1_1 \in \mathcal{P}_1$ for $i \in [p_1]$ such that

$$g^1_1 = w^i_1 \in L^2(S_1, \mathbb{R}^{p_0}) = L^2(S_1, \mathcal{H}_0) = \mathcal{P}_1, \quad f^1_1(z) = ||z|| \sigma(\langle g^i_1, z \rangle / ||z||) \text{ for } z \in \mathcal{P}_1.$$ 

Based on layer $k-1$, the forward function $f^i_k \in \mathcal{H}_k$ and $g^i_k \in \mathcal{P}_k$ for channel $i \in [p_k]$ at layer $k$ are

$$g^i_k(v) = \sum_{j=1}^{p_{k-1}} w^j_{k-1,i} f^j_{k-1} \text{ for } v \in S_k, \quad f^i_k(z) = ||z|| \sigma(\langle g^i_k, z \rangle / ||z||) \text{ for } z \in \mathcal{P}_k.$$ 

And the output layer sets $g_\sigma(u) = \sum_{j=1}^{p_n} w^j_{n+1}(u) f^j_n$ for all $u \in \Omega$, so that $f : x_0 \mapsto \langle g_\sigma, x_n \rangle$ exactly recovers $f_\sigma$ as shown by [29].

Effect of warping in CKN. Since warping does not change the set of functions in the RKHS at each layer, the CKN constructed above is obviously retained in our new space of CKNs. However, interesting changes occur to the RKHS norm. As shown by [29], $\|f^i_k\|_{\mathcal{H}_k}^2 \leq C_\sigma^2(\|w^i_k\|_{H_k}^2)$ where $\|w^i_k\|_{H_k}^2 = \int_{S_1} \|w^i_k(v)\|^2 \, dv_1(v)$, and $C_\sigma : \mathbb{R} \rightarrow \mathbb{R}$ is an increasing function depending only on $\sigma$ and the kernels. Recursively, without kernel warping, we have

$$\|f^i_k\|_{\mathcal{H}_k}^2 \leq C_\sigma^2(\|g^i_k\|_{H_{k-1}}^2), \quad \|g^i_k\|_{H_{k-1}}^2 \leq \sum_{j=1}^{p_{k-1}} \|w^j_{k-1,i}\|^2_2 \cdot \|f^j_{k-1}\|_{H_{k-1}}^2.$$ 

(70)

So the overall recursion on $\|f^i_k\|_{\mathcal{H}_k}^2$ writes

$$\|f^i_k\|_{\mathcal{H}_k}^2 \leq C_\sigma^2 \left( \sum_{j=1}^{p_{k-1}} \|w^j_{k-1,i}\|^2_2 \cdot \|f^j_{k-1}\|_{H_{k-1}}^2 \right).$$ 

(71)

And the final prediction $f \in L^2(\Omega, \mathcal{H}_n)$ can be bounded by

$$\|f\|_{\mathcal{H}_n}^2 \leq p_n \sum_{j=1}^{p_n} \left[ \int_{\Omega} |w^j_{n+1}(u)|^2 \, du \right] \|f^j_n\|_{H_n}^2.$$ 

(72)

With the same functions $f^i_k$ and $g^i_k$, we note that the RKHS norm of $f$ can only increase because $\|f\|_{\mathcal{H}_n}^2 = \|f\|_{\mathcal{H}_n}^2 + \sum_{i=1}^{m} \langle z_i, f \rangle_{\mathcal{H}_n}^2$, if we warp the kernel and RKHS of the last CKN layer (layer $n$, before the output layer). We note in passing that this argument can be applied only to the last layer because warping is applied to images rather than patches, while the recursion in (70) and (71) works only on patches. We leave it as future work to show that warping an image at intermediate layers will also keep or increase the RKHS norm on all patches.