A Stein–Papangelou Goodness-of-Fit Test for Point Processes (Appendix)

A Omitted Proofs and Results

Proof of Theorem 5. By the reproducing property of $\mathcal{H}$, $h(\phi) = \langle h(\cdot), k(\phi, \cdot)\rangle_{\mathcal{H}_k}$. For any $x \in \mathbb{X}$, we have

$$(\mathcal{D}_x^+ h)(\phi) = \langle h(\cdot), k(\phi + \delta_x, \cdot)\rangle_{\mathcal{H}_k} - \langle h(\cdot), k(\phi, \cdot)\rangle_{\mathcal{H}_k} = \left\langle h(\cdot), (\mathcal{D}_x^+ k)(\phi, \cdot)\right\rangle_{\mathcal{H}_k};$$
$$(\mathcal{D}_x^- h)(\phi) = \langle h(\cdot), k(\phi, \cdot)\rangle_{\mathcal{H}_k} - \langle h(\cdot), k(\phi - \delta_x, \cdot)\rangle_{\mathcal{H}_k} = \left\langle h(\cdot), (\mathcal{D}_x^- k)(\phi, \cdot)\right\rangle_{\mathcal{H}_k}.$$

Thus, by Eq. (7),

$$\begin{aligned}
\mathbb{E}_{\Phi\sim\eta}\left[\mathcal{A}_\rho h(\Phi)\right] &= \mathbb{E}_{\Phi\sim\eta}\left[\int_{\mathbb{X}} (\mathcal{D}_x^+ h)(\Phi)\rho(x|\Phi)\,\mathrm{d}x - \int_{y\in\mathbb{X}} (\mathcal{D}_x^- h)(\Phi)\Phi(\mathrm{d}x)\right] \\
&= \mathbb{E}_{\Phi\sim\eta}\left[\int_{\mathbb{X}} \left\langle h(\cdot), (\mathcal{D}_x^+ k)(\Phi, \cdot)\right\rangle_{\mathcal{H}_k} \rho(x|\Phi)\,\mathrm{d}x - \int_{y\in\mathbb{X}} \left\langle h(\cdot), (\mathcal{D}_x^- k)(\Phi, \cdot)\right\rangle_{\mathcal{H}_k} \Phi(\mathrm{d}x)\right] \\
&= \mathbb{E}_{\Phi\sim\eta}\left[\left\langle h(\cdot), \int_{\mathbb{X}} (\mathcal{D}_x^+ k)(\Phi, \cdot)\,\rho(x|\Phi)\,\mathrm{d}x\right\rangle_{\mathcal{H}_k} - \left\langle h(\cdot), \int_{\mathbb{X}} (\mathcal{D}_x^- k)(\Phi, \cdot)\,\Phi(\mathrm{d}x)\right\rangle_{\mathcal{H}_k}\right] \\
&= \langle h(\cdot), \mathbb{E}_{\Phi\sim\eta}\left[\mathcal{A}_\rho k(\Phi, \cdot)\right]\rangle_{\mathcal{H}_k}.
\end{aligned}$$

Defining $\beta_{\eta,\rho} := \mathbb{E}_{\Phi\sim\eta}\left[\mathcal{A}_\rho k(\Phi, \cdot)\right]$, we can rewrite the kernelized Stein discrepancy as

$$\mathbb{D}_{\mathcal{H}_k}(\eta \,\|\, \rho) = \sup_{h\in\mathcal{H},\, \|h\|_{\mathcal{H}_k}\le 1} \langle h, \beta_{\eta,\rho}\rangle_{\mathcal{H}_k},$$

which immediately implies that $\mathbb{D}_{\mathcal{H}_k}(\eta \,\|\, \rho) = \|\beta_{\eta,\rho}\|_{\mathcal{H}_k}$ since the supremum will be obtained by $h = \beta_{\eta,\rho}/\|\beta_{\eta,\rho}\|_{\mathcal{H}_k}$. Therefore, we can write

$$\begin{aligned}
\mathbb{D}^2_{\mathcal{H}_k}(\eta \,\|\, \rho) = \|\beta_{\eta,\rho}\|^2_{\mathcal{H}_k} &= \left\langle \mathbb{E}_{\Phi\sim\eta}\left[\mathcal{A}_\rho^\Phi k(\Phi, \cdot)\right], \mathbb{E}_{\Psi\sim\eta}\left[\mathcal{A}_\rho^\Psi k(\Psi, \cdot)\right]\right\rangle_{\mathcal{H}_k} = \mathbb{E}_{\Phi,\Psi\sim\eta}\left[\left\langle \mathcal{A}_\rho^\Phi k(\Phi, \cdot), \mathcal{A}_\rho^\Psi k(\Psi, \cdot)\right\rangle_{\mathcal{H}_k}\right] \\
&= \mathbb{E}_{\Phi,\Psi\sim\eta}\left[\mathcal{A}_\rho^\Phi \mathcal{A}_\rho^\Psi \langle k(\Phi, \cdot), k(\Psi, \cdot)\rangle_{\mathcal{H}_k}\right] = \mathbb{E}_{\Phi,\Psi\sim\eta}\left[\mathcal{A}_\rho^\Phi \mathcal{A}_\rho^\Psi k(\Phi, \Psi)\right],
\end{aligned}$$

where we applied the reproducing property, $\langle k(\Phi, \cdot), k(\Psi, \cdot)\rangle_{\mathcal{H}_k} = k(\Phi, \Psi)$.

Deriving the expression in Eq. (12). Fixing ψ and applying Eq. (7) to $k(\phi, \psi)$ viewed as a function of ϕ, we have

$$\mathcal{A}_\rho^\phi k(\phi, \psi) = \int_{\mathbb{X}} \left[k(\phi + \delta_u, \psi) - k(\phi, \psi)\right] \rho(u|\phi)\,\mathrm{d}u + \sum_{x\in\phi} \left[k(\phi - \delta_x, \psi) - k(\phi, \psi)\right].$$

Now, fixing ϕ and applying Eq. (7) to $\mathcal{A}_\rho^\phi k(\phi, \psi)$ viewed as a function of ψ, we have

$$\begin{aligned}
\mathcal{A}_\rho^\psi \mathcal{A}_\rho^\phi k(\phi, \psi) &= \int_{\mathbb{X}} \left[\mathcal{A}_\rho^\phi k(\phi, \psi + \delta_v) - \mathcal{A}_\rho^\phi k(\phi, \psi)\right] \rho(v|\psi)\,\mathrm{d}v + \sum_{y\in\psi} \left[\mathcal{A}_\rho^\phi k(\phi, \psi - \delta_y) - \mathcal{A}_\rho^\phi k(\phi, \psi)\right] \\
&= \int_{\mathbb{X}} \Bigg[\left(\int_{\mathbb{X}} \left[k(\phi + \delta_u, \psi + \delta_v) - k(\phi, \psi + \delta_v)\right] \rho(u|\phi)\,\mathrm{d}u + \sum_{x\in\phi} \left[k(\phi - \delta_x, \psi + \delta_v) - k(\phi, \psi + \delta_v)\right]\right) \\
&\qquad - \left(\int_{\mathbb{X}} \left[k(\phi + \delta_u, \psi) - k(\phi, \psi)\right] \rho(u|\phi)\,\mathrm{d}u + \sum_{x\in\phi} \left[k(\phi - \delta_x, \psi) - k(\phi, \psi)\right]\right)\Bigg] \rho(v|\psi)\,\mathrm{d}v
\end{aligned}$$

$$
\begin{aligned}
&+\sum_{y\in\psi}\Bigg[\Bigg(\int_{\mathbb{X}}\left[k(\phi+\delta_u,\psi-\delta_y)-k(\phi,\psi-\delta_y)\right]\rho(u|\phi)\,\mathrm{d}u+\sum_{x\in\phi}\left[k(\phi-\delta_x,\psi-\delta_y)-k(\phi,\psi-\delta_y)\right]\Bigg)\\
&\qquad-\Bigg(\int_{\mathbb{X}}\left[k(\phi+\delta_u,\psi)-k(\phi,\psi)\right]\rho(u|\phi)\,\mathrm{d}u+\sum_{x\in\phi}\left[k(\phi-\delta_x,\psi)-k(\phi,\psi)\right]\Bigg)\Bigg]\\
=&\int_{\mathbb{X}}\int_{\mathbb{X}}\Big[k(\phi+\delta_u,\psi+\delta_v)-k(\phi,\psi+\delta_v)-k(\phi+\delta_u,\psi)+k(\phi,\psi)\Big]\rho(u|\phi)\,\rho(v|\psi)\,\mathrm{d}u\mathrm{d}v\\
&+\int_{\mathbb{X}}\Bigg[\sum_{x\in\phi}\big[k(\phi-\delta_x,\psi+\delta_v)-k(\phi-\delta_x,\psi)\big]-|\phi|\cdot\big[k(\phi,\psi+\delta_v)-k(\phi,\psi)\big]\Bigg]\rho(v|\psi)\,\mathrm{d}v\\
&+\int_{\mathbb{X}}\Bigg[\sum_{y\in\psi}\big[k(\phi+\delta_u,\psi-\delta_y)-k(\phi,\psi-\delta_y)\big]-|\psi|\cdot\big[k(\phi+\delta_u,\psi)-k(\phi,\psi)\big]\Bigg]\rho(u|\phi)\,\mathrm{d}u\\
&+\Bigg[\sum_{x\in\phi}\sum_{y\in\psi}k(\phi-\delta_x,\psi-\delta_y)-|\phi|\cdot\sum_{y\in\psi}k(\phi,\psi-\delta_y)-|\psi|\cdot\sum_{x\in\phi}k(\phi-\delta_x,\psi)+|\phi|\cdot|\psi|\cdot k(\phi,\psi)\Bigg],
\end{aligned}
$$

which recovers the expression in Eq. (12). This concludes the proof of Theorem 5. □

Theorem 7 (Adapted from [30])**.** *Let $k(\cdot,\cdot)$ be a positive definite kernel on $\mathcal{N}_{\mathbb{X}}$, and assume that $\mathbb{E}_{\Phi,\Psi\sim\eta}\left[\kappa_\rho(\Phi,\Psi)^2\right]<\infty$. We have the following two cases:*

(i) If $\eta\neq\rho$, then $\widehat{\mathbb{S}}(\eta\,\|\,\rho)$ is asymptotically normal:

$$\sqrt{m}\left(\widehat{\mathbb{S}}(\eta\,\|\,\rho)-\mathbb{S}(\eta\,\|\,\rho)\right)\xrightarrow{\mathcal{D}}\mathcal{N}(0,\sigma^2),$$

where $\sigma^2=\mathrm{Var}_{\Phi\sim\eta}(\mathbb{E}_{\Psi\sim\eta}\left[\kappa_\rho(\Phi,\Psi)\right])>0$.

(ii) If $\eta=\rho$, then $\sigma^2=0$, and the U-statistic is degenerate.

Lemma 8 (Schoenberg [37])**.** *The function*

$$k(x,y):=\exp\left\{-\frac{f(x,y)}{\ell}\right\}$$

defined on a domain $\mathcal{D}$ is a positive definite kernel for all $\ell>0$ if and only if f is a conditionally negative definite *function, i.e., $\sum_{i,j=1}^n c_ic_jf(x_i,x_j)\le 0$ for all $n\in\mathbb{N}$, $x_1,\dots,x_n\in\mathcal{D}$, and $c_1,\dots,c_n\in\mathbb{R}$ such that $\sum_{i=1}^n c_i=0$.*

Proof of Proposition 6. Denote

$$\xi(\phi,\psi):=\sum_{x\in\phi}\sum_{y\in\psi}k_{\mathbb{X}}(x,y).\tag{17}$$

By Proposition 3.1 of [17], $\xi(\cdot,\cdot)$ is a p.d. kernel on $\mathcal{N}_{\mathbb{X}}$ if $k_{\mathbb{X}}$ is a p.d. kernel on $\mathbb{X}$. By Lemma 8, to show that Eq. (15) defines a p.d. kernel, it suffices to show that $\widehat{d}^2$ is a conditionally negative-definite function. To this end, observe that for any $n\in\mathbb{N}$, $\phi_1,\dots,\phi_n\in\mathcal{N}_{\mathbb{X}}$, and $c_1,\dots,c_n\in\mathbb{R}$ satisfying $\sum_{i=1}^n c_i=0$, we have

$$
\begin{aligned}
\sum_{i=1}^n\sum_{j=1}^n c_i\,c_j\,\widehat{d}^2(\phi_i,\phi_j)=&\frac{1}{|\phi|^2}\left(\sum_{i=1}^n c_i\sum_{x\in\phi_i}\sum_{x'\in\phi_i}k_{\mathbb{X}}(x,x')\right)\left(\sum_{j=1}^n c_j\right)+\frac{1}{|\psi|^2}\left(\sum_{i=1}^n c_i\right)\left(\sum_{j=1}^n c_j\sum_{y\in\phi_j}\sum_{y'\in\phi_j}k_{\mathbb{X}}(y,y')\right)\\
&-\frac{2}{|\phi_i|\cdot|\phi_j|}\sum_{i=1}^n\sum_{j=1}^n c_i\,c_j\,\xi(\phi_i,\phi_j)\\
=&-\frac{2}{|\phi_i|\cdot|\phi_j|}\sum_{i=1}^n\sum_{j=1}^n c_i\,c_j\,\xi(\phi_i,\phi_j)\\
\le&\ 0\,,
\end{aligned}
$$

where we used the fact that $\xi(\cdot,\cdot)$ is a p.d. kernel on $\mathcal{N}_{\mathbb{X}}$. Thus, the proof is complete. □

B KSD Goodness-of-Fit Testing Algorithm

The KSD goodness-of-fit testing procedure is summarized in Algorithm 1.

Algorithm 1 KSD goodness-of-fit test for point processes

1: **Input:** Papangelou conditional intensity ρ, point configurations $\{\mathcal{X}_i\}_{i=1}^m \sim \eta$, kernel function $k(\cdot,\cdot)$ on $\mathcal{N}_{\mathbb{X}}$, bootstrap sample size $\widetilde{m}$, significance level α.
2: **Objective:** Test $H_0 : \rho = \eta$ vs. $H_1 : \rho \neq \eta$.
3: Compute test statistic $\widehat{\mathbb{S}}\,(\eta \,\|\, \rho)$ via

$$\widehat{\mathbb{S}}\,(\eta \,\|\, \rho) = \frac{1}{m(m-1)} \sum_{i=1}^{m} \sum_{j \neq i}^{m} \kappa_\rho(\mathcal{X}_i, \mathcal{X}_j),$$

where

$$\begin{aligned}\kappa_\rho(\phi,\psi) = &\int_{\mathbb{X}}\int_{\mathbb{X}} \Big[k(\phi\cup\{u\},\psi\cup\{v\}) - k(\phi,\psi\cup\{v\}) - k(\phi\cup\{u\},\psi) + k(\phi,\psi)\Big]\rho(u|\phi)\,\rho(v|\psi)\,\mathrm{d}u\,\mathrm{d}v \\ &+ \int_{\mathbb{X}} \Bigg[\sum_{x\in\phi}\big[k(\phi\backslash\{x\},\psi\cup\{v\}) - k(\phi\backslash\{x\},\psi)\big] - |\phi|\cdot\big[k(\phi,\psi\cup\{v\}) - k(\phi,\psi)\big]\Bigg]\rho(v|\psi)\,\mathrm{d}v \\ &+ \int_{\mathbb{X}} \Bigg[\sum_{y\in\psi}\big[k(\phi\cup\{u\},\psi\backslash\{y\}) - k(\phi,\psi\backslash\{y\})\big] - |\psi|\cdot\big[k(\phi\cup\{u\},\psi) - k(\phi,\psi)\big]\Bigg]\rho(u|\phi)\,\mathrm{d}u \\ &+ \Bigg[\sum_{x\in\phi}\sum_{y\in\psi} k(\phi\backslash\{x\},\psi\backslash\{y\}) - |\phi|\cdot\sum_{y\in\psi} k(\phi,\psi\backslash\{y\}) - |\psi|\cdot\sum_{x\in\phi} k(\phi\backslash\{x\},\psi) + |\phi|\cdot|\psi|\cdot k(\phi,\psi)\Bigg].\end{aligned}$$

4: **for** $b = 1, \ldots, \widetilde{m}$ **do**
5: Draw random multinomial weights $w_1, \ldots, w_m \sim \mathrm{Mult}(m; {}^1\!/_m, \ldots, {}^1\!/_m)$; set $\widetilde{w}_i = (w_i - 1)/m$.
6: Compute bootstrap test statistic $\widehat{\mathbb{S}}_b^*$ via

$$\widehat{\mathbb{S}}_b^*(\eta \,\|\, \rho) = \sum_{i=1}^{m} \sum_{j \neq i}^{m} \widetilde{w}_i \widetilde{w}_j \kappa_p(\mathcal{X}_i, \mathcal{X}_j).$$

7: Compute critical value $\gamma_{1-\alpha}$ by taking the $(1-\alpha)$-th quantile of the bootstrapped statistics $\{\widehat{\mathbb{S}}_b^*\}_{b=1}^{\widetilde{m}}$.
8: **Output:** Reject H_0 if $\widehat{\mathbb{S}}\,(\eta \,\|\, \rho) > \gamma_{1-\alpha}$, otherwise do not reject H_0.

For concreteness, we provide an example Python implementation of Eq. (12) below:

```python
def kernel(X, Y):
    """
    Evaluates a kernel function for two point-sets X and Y.

    Args:
        X, Y: numpy arrays of shape (_, d), collections of d-dimensional points.

    Returns:
        float, value of k(X, Y).
    """

def papangelou(u, X):
    """
    Evaluates the Papangelou conditional intensity (u|X) of a point process
        at location u given observed point-set X.

    Args:
        u: numpy array of shape (d,), a location in the ground space.
        X: numpy array of shape (n, d), the given point-set.

    Returns:
        float, value of rho(u|X).
    """
```

```python
def integrate(func, domain):
    """
    Integrates a (univariate or multivariate) function func over domain.
        See scipy.integrate for a list of common numerical integration routines.

    Args:
        func: function, a univariate or multivariate function.
        domain: list of lists, the integration ranges for each variable.

    Returns:
        float, value of the definite integral.
    """

def kappa(X, Y, domain):
    """
    Evaluates Eq.(12) for two point-sets X and Y using kernel() and papangelou().

    Args:
        X, Y: numpy arrays of shape (_, d), collections of d-dimensional points.
        domain: list of lists, ranges specifying the ground space.

    Returns:
        float, value of kappa(X, Y).
    """
    n = X.shape[0]
    m = Y.shape[0]

    k = kernel(X, Y)
    k_X = sum(kernel(np.delete(X, i, axis=0), Y) for i in xrange(n))
    k_Y = sum(kernel(X, np.delete(Y, j, axis=0)) for j in xrange(m))
    k_X_Y = sum(kernel(np.delete(X, i, axis=0), np.delete(Y, j, axis=0))
                for i in xrange(n) for j in xrange(m))

    def integrand_uv(u, v):
        # Double integrand over u and v
        k_uv = kernel(np.vstack((X, u)), np.vstack((Y, v)))
        k_v = kernel(X, np.vstack((Y, v)))
        k_u = kernel(np.vstack((X, u)), Y)
        c_u = papangelou(u, X)
        c_v = papangelou(v, Y)
        return (k_uv - k_v - k_u + k) * c_u * c_v

    def integrand_v(v):
        # Integrand over v
        k_X_v = sum(kernel(np.delete(X, i, axis=0), np.vstack((Y, v)))
                    for i in xrange(n))
        k_v = kernel(X, np.vstack((Y, v)))
        c_v = papangelou(v, Y)
        return ((k_X_v - k_X) - n*(k_v - k)) * c_v

    def integrand_u(u):
        # Integrand over u
        k_Y_u = sum(kernel(np.vstack((X, u)), np.delete(Y, j, axis=0))
                    for j in xrange(m))
        k_u = kernel(np.vstack((X, u)), Y)
        c_u = papangelou(u, X)
        return ((k_Y_u - k_Y) - m*(k_u - k)) * c_u

    # Compute integrals
    term1 = integrate(integrand_uv, domain)
    term2 = integrate(integrand_v, domain)
    term3 = integrate(integrand_u, domain)
    term4 = k_X_Y - n*k_Y - m*k_X + m*n*k

    return term1 + term2 + term3 + term4
```