PsiFormer: Una arquitectura Transformer para la mecánica cuántica de muchos cuerpos

Jorge Alvaro Munoz Laredo[1]

[1]Facultad de Física, Universidad Nacional de Ingeniería, Lima, Perú, jorge.munoz.l@uni.pe

December 15, 2025

Abstract

Con soluciones precisas de la ecuación de Schrödinger de muchos electrones, toda la química podría derivarse desde primeros principios, pero el tratamiento analítico es intratable debido a las correlaciones electrón-electrón intrínsecamente fuertes, la antisimetría y el comportamiento cuspídeo. Recientemente, gracias a su alta flexibilidad, se han aplicado enfoques de aprendizaje profundo a este problema; modelos de funciones de onda neuronales como FermiNet y PauliNet han avanzado en precisión, pero el costo computacional y el error típicamente crecen de forma pronunciada con el tamaño del sistema, lo que limita la aplicabilidad a moléculas más grandes. Además, carecen de arquitecturas sólidas diseñadas para capturar correlaciones electrónicas de largo alcance con atención escalable. En este trabajo desarrollo Psiformer, un ansatz basado en Transformers que acopla atención escalable con una estructura consciente de la física. El entrenamiento se formula dentro de Variational Monte Carlo (VMC); la evaluación se realizará comparando la energía del estado fundamental frente a otros métodos tradicionales. También expongo preguntas de diseño para mejoras futuras, incluyendo atención dispersa/global y elecciones de optimizador inspiradas por avances recientes en Transformers.

1 Introducción

El problema de la estructura electrónica sigue siendo desafiante: las funciones de onda, que describen completamente el sistema, viven en un espacio de dimensión $3N$, donde N es el número de electrones; cada uno vive en el espacio tridimensional. Además, debe satisfacer propiedades específicas impuestas por las leyes físicas y vivir en el espacio complejo.

Aunque las leyes que lo gobiernan se conocen desde hace casi un siglo [1], obtener aproximaciones prácticas a la función de onda cuántica de muchos cuerpos sigue siendo difícil. Enfoques establecidos como la teoría del funcional de la densidad [2], Born Oppenheimer [3], y ansätze variacionales estructurados [4] sacrifican generalidad para ganar tratabilidad al imponer formas funcionales específicas o aproximaciones a la correlación. Estas elecciones son efectivas dentro de sus regímenes, pero pueden tener dificultades en sistemas fuertemente correlacionados o escalar mal con el número de electrones cite. Aquí es donde entran los métodos modernos basados en aprendizaje: en lugar de fijar la forma funcional, la aprendemos, mientras imponemos la física esencial (antisimetría, comportamiento cuspídeo, simetría por permutación); aquí es donde brillan los métodos de aprendizaje profundo. Este campo ha reconfigurado varios dominios científicos, desde la predicción de estructura de proteínas [5] hasta el modelado en visión [6] y sustitutos de EDP [7]. Motivada por estos éxitos, la comunidad ha explorado enfoques neuronales para problemas cuánticos de muchos cuerpos, buscando aproximaciones precisas y escalables a la función de onda de muchos electrones [8, 9].

Así, los modelos de función de onda neuronal han surgido como una alternativa prometedora. Arquitecturas como **FermiNet** [10] y **PauliNet** [11] combinan aproximadores flexibles con estructuras determinantes para respetar la antisimetría, mejorando la expresividad variacional. Sin embargo, persisten dos limitaciones prácticas. Primero, el error o el costo de cómputo a menudo escalan de forma desfavorable con el número de electrones, restringiendo la aplicabilidad a moléculas más grandes. Segundo, los mecanismos de **correlación electrónica de largo alcance**, centrales para los efectos de Coulomb e intercambio, suelen ser implícitos o costosos de capturar, lo que conduce a dificultades de optimización y a una generalización frágil.

Los Transformers ofrecen una dirección atractiva. La autoatención proporciona interacciones directas de muchos-a-muchos entre tokens en una sola capa, es altamente paralelizable y ha mostrado un escalamiento favorable en otros dominios (procesamiento de lenguaje natural [12]). Para la estructura electrónica, donde cualquier electrón puede interactuar con cualquier otro y los índices de electrones son intercambiables, la atención se alinea de forma natural con la física: permite acoplamiento global sin imponer un orden arbitrario. El reto es incorporar los **sesgos inductivos** correctos (conciencia de distancia, estructura de espín, tratamiento de cúspides) y mantener la **antisimetría fermiónica** mientras se controla el costo computacional.

En este trabajo desarrollo **Psiformer**, un ansatz variacional basado en Transformers para sistemas de muchos electrones [13]. Psiformer usa autoatención para construir características ricas por electrón, informadas por descriptores electrón–electrón y electrón–núcleo, y luego impone antisimetría explícitamente mediante cabezas basadas en determinantes. Se incorporan priors conscientes de la física (por ejemplo, codificaciones de distancia/radiales y embeddings motivados por cúspides) para reducir la complejidad de muestreo y estabilizar el entrenamiento. La optimización se formula dentro de **Variational Monte Carlo (VMC)** [4] minimizando la energía variacional.

Contribuciones.

- Una función de onda neuronal basada en Transformers (**Psiformer**) [13] implementada en PyTorch [14] que separa el modelado de correlación (atención) de la simetría fermiónica (cabezas determinantes) mientras incorpora priors conscientes de Coulomb.
- Una receta de entrenamiento VMC con elecciones prácticas para la estabilidad (diseño de características y preacondicionamiento opcional por gradiente natural).

Los objetivos son los siguientes:

2 Objetivos

- Obtener un modelo capaz de aproximar la energía del estado fundamental para átomos y moléculas específicas.
- Comparar nuestro modelo con otros métodos de estado del arte para resolver la ecuación de Schrödinger de muchos electrones.
- Buscar mejoras futuras al intentar abordar moléculas más grandes.

3 Visión general

Este trabajo está estructurado de la siguiente manera: el marco teórico section 4 introduce los fundamentos de la teoría cuántica de muchos cuerpos, la estructura de la ecuación de Schrödinger para muchos cuerpos, así como conceptos fundamentales de aprendizaje profundo que se usarán en el desarrollo de este problema específico; section 5 introduce **Psi Former**, una arquitectura basada en Transformers construida sobre **Fermi Net**; y section 6 especifica las herramientas y entornos usados para implementar este trabajo.

4 Marco teórico

La ecuación de Schrödinger fue presentada en una serie de publicaciones realizadas por Erwin Schrödinger en el año 1926 [1]. Allí buscamos la función compleja ψ, que vive en un espacio de Hilbert $\mathcal{H}$ y se denomina **función de onda**. Para una partícula única, no relativista y sin espín, esta función depende de la posición de la partícula $\tilde{\mathbf{r}}$ y del tiempo t ($\psi(\tilde{\mathbf{r}}, t)$). La cantidad $|\psi(\mathbf{r}, t)|^2$ es la **densidad de probabilidad** de encontrar la partícula cerca de $\mathbf{r}$ en el tiempo t [15].

Guiado por el descubrimiento de de Broglie [16] de la dualidad onda–partícula y por una intuición muy aguda, Schrödinger propuso la ecuación dependiente del tiempo (TDSE):

$$i\hbar\frac{\partial\psi}{\partial t} = \hat{H}\psi \tag{1}$$

donde i es la unidad imaginaria, $\hbar$ es la constante de Planck reducida, aproximadamente $1.054571817\cdots\times 10^{-34}\,J\cdot s$, y $\hat{H}$ es un operador lineal hermítico llamado el **Hamiltoniano**, que representa la energía total del sistema. Para una partícula única no relativista (de baja energía) de masa m en un potencial escalar $V(\mathbf{r}, t)$, toma la forma:

$$\hat{H} = \frac{\hat{\mathbf{p}}^2}{2m} + V(\mathbf{r}, t) \tag{2}$$

donde $\hat{\mathbf{p}}$ es el operador momento y, en la **representación de posición**, toma la forma [17]:

$$\hat{\mathbf{p}} = -i\hbar\nabla \tag{3}$$

donde ∇ es el operador laplaciano; por lo tanto, la ecuación de Schrödinger dependiente del tiempo (TDSE) se escribe explícitamente como:

$$i\hbar\,\frac{\partial\psi}{\partial t} = \left[-\frac{\hbar^2}{2m}\nabla^2 + V(\mathbf{r}, t)\right]\psi \tag{4}$$

La **forma independiente del tiempo** (TISE) puede derivarse a partir de la ecuación 1 cuando la función de onda ψ puede escribirse como el producto de dos funciones R y T, donde R depende únicamente de la parte espacial ($\mathbf{r}$) y T únicamente de la parte temporal (t), esto es:

$$\psi(\mathbf{r}, t) = R(\mathbf{r})T(t) \tag{5}$$

Sustituyendo esta forma en la ecuación 1, se puede derivar [17] que:

$$T(t) = e^{-iEt/\hbar}$$

donde E, la energía del sistema, es una constante. También se obtiene que la parte espacial está determinada por:

$$\hat{H}R(\mathbf{r}) = ER(\mathbf{r}) \tag{6}$$

Vamos a representar la función espacial R como ψ. En este trabajo nos enfocaremos únicamente en la TISE: cuando tratamos con energía constante, los electrones se encuentran casi siempre cerca del estado de menor energía, conocido como el estado fundamental. Las soluciones con mayor energía, conocidas como estados excitados, son relevantes en fotoquímica, pero en este trabajo restringiremos nuestra atención a estados fundamentales.

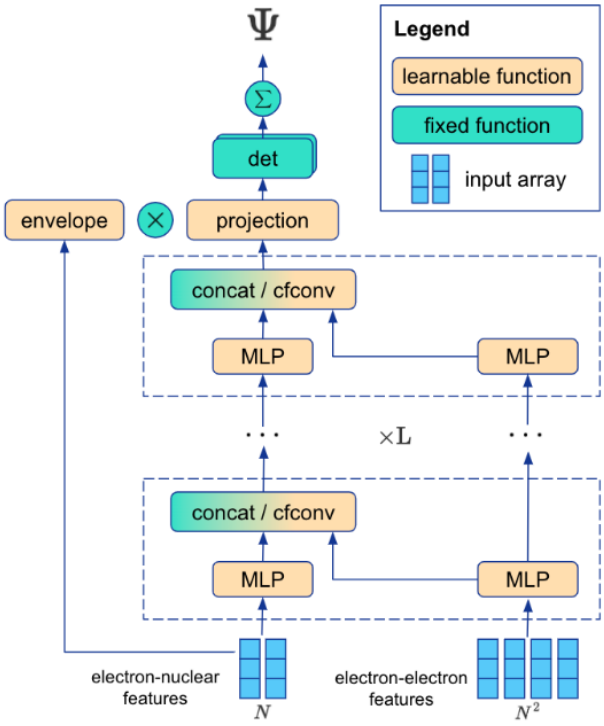

Figure 1: FermiNet tiene dos flujos, que actúan sobre características electrón–núcleo y electrón–electrón, los cuales se combinan mediante concatenación; imagen tomada de [13].

4.0.1 La ecuación de Schrödinger de muchos electrones

Cuando consideramos más de una sola partícula, incluimos el espín (σ) y la interacción entre partículas. Así, en su forma independiente del tiempo, la ecuación de Schrödinger puede escribirse como un problema de valores propios:

$$\hat{H}\psi(\mathbf{x}_1,\ldots,\mathbf{x}_n) = E\psi(\mathbf{x}_1,\ldots,\mathbf{x}_n) \tag{7}$$

donde $\mathbf{x}_i = \{\mathbf{r}_i, \sigma\}$, $\mathbf{r}_i \in \mathbb{R}^3$ es la posición de cada electrón y $\sigma \in \{\uparrow,\downarrow\}$ es el llamado espín. Para modelar la energía potencial de un sistema de muchos cuerpos (p. ej., átomos, moléculas), primero debemos considerar el potencial dado por la repulsión entre electrones:

$$V_{ij} = \frac{e^2}{4\pi\varepsilon_0}\frac{1}{|\mathbf{r}_i - \mathbf{r}_j|} \tag{8}$$

Aquí $e = 1.602\,176\,634\times10^{-19}$ C es la carga elemental, $\varepsilon_0 = 8.854\,187\,8128 \times 10^{-12}$ F m^{-1} es la permitividad eléctrica del vacío, y $\mathbf{r}_i$ es el vector posición del electrón i en el sistema de referencia elegido. El potencial debido a la atracción entre el protón I y el electrón i está dado por:

$$V_{iI} = -\frac{1}{4\pi\varepsilon_0}\frac{eZ_I}{|\mathbf{r}_i - \mathbf{R}_I|} \tag{9}$$

donde Z_I es el número atómico del núcleo I (por ejemplo, en un átomo de helio $Z = 2$) y $\mathbf{R}_I$ es la posición de dicho núcleo en el sistema de referencia elegido.
El sistema de referencia usualmente se toma en el **centro de masa** o en el **centro de la molécula**. El potencial dado por la repulsión entre los núcleos I y J (protones) es:

$$V_{IJ} = \frac{1}{4\pi\varepsilon_0}\frac{Z_I Z_J}{|\mathbf{R}_I - \mathbf{R}_J|} \tag{10}$$

donde ∇ es el operador laplaciano; por lo tanto, la ecuación de Schrödinger dependiente del tiempo (TDSE) se escribe explícitamente como:

$$i\hbar\frac{\partial\psi}{\partial t} = \left[-\frac{\hbar^2}{2m}\nabla^2 + V(\mathbf{r},t)\right]\psi \tag{11}$$

La **forma independiente del tiempo** (TISE) puede derivarse a partir de la ecuación 1 cuando la función de onda ψ puede escribirse como el producto de dos funciones R y T, donde R depende únicamente de la parte espacial ($\mathbf{r}$) y T únicamente de la parte temporal (t), esto es:

$$\psi(\mathbf{r},t) = R(\mathbf{r})T(t) \tag{12}$$

Sustituyendo esta forma en la ecuación 1, se puede derivar [17] que:

$$T(t) = e^{-iEt/\hbar}$$

donde E, la energía del sistema, es una constante. También se obtiene que la parte espacial está determinada por:

$$\hat{H}R(\mathbf{r}) = ER(\mathbf{r}) \tag{13}$$

En adelante, representaremos la función espacial R como ψ. En este trabajo nos enfocaremos únicamente en la TISE: cuando tratamos con energía constante, los electrones se encuentran casi siempre cerca del estado

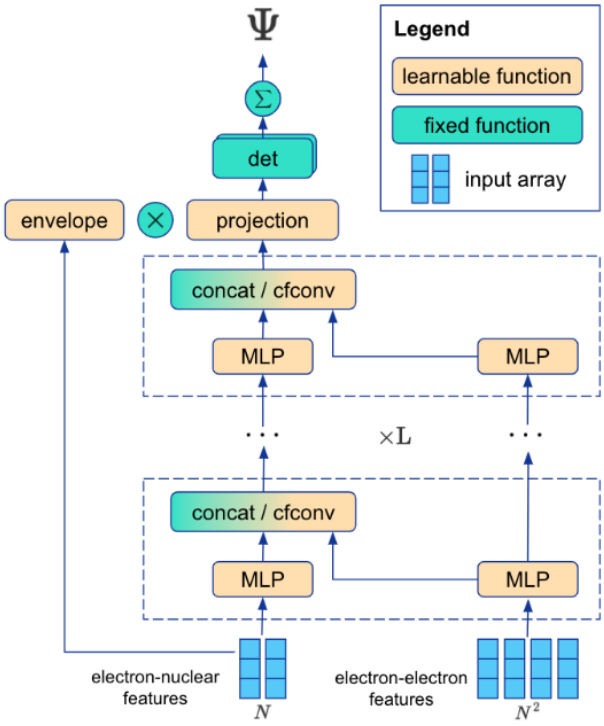

Figure 2: FermiNet tiene dos flujos, que actúan sobre características electrón–núcleo y electrón–electrón, los cuales se combinan mediante concatenación; imagen tomada de [13].

de menor energía, conocido como el estado fundamental. Las soluciones con mayor energía, conocidas como estados excitados, son relevantes en fotoquímica, pero en este trabajo restringiremos nuestra atención a estados fundamentales.

4.0.2 La ecuación de Schrödinger de muchos electrones

Cuando consideramos más de una sola partícula, incluimos el espín (σ) y la interacción entre partículas. Así, en su forma independiente del tiempo, la ecuación de Schrödinger puede escribirse como un problema de valores propios:

$$\hat{H}\psi(\mathbf{x}_1,\ldots,\mathbf{x}_n) = E\psi(\mathbf{x}_1,\ldots,\mathbf{x}_n) \tag{14}$$

donde $\mathbf{x}_i = \{\mathbf{r}_i, \sigma\}$, $\mathbf{r}_i \in \mathbb{R}^3$ es la posición de cada electrón y $\sigma \in \{\uparrow,\downarrow\}$ es el llamado espín. Para modelar la energía potencial de un sistema de muchos cuerpos (p. ej., átomos, moléculas), primero debemos considerar el potencial dado por la repulsión entre electrones:

$$V_{ij} = \frac{e^2}{4\pi\varepsilon_0}\frac{1}{|\mathbf{r}_i - \mathbf{r}_j|} \tag{15}$$

Aquí $e = 1.602\,176\,634\times10^{-19}$ C es la carga elemental, $\varepsilon_0 = 8.854\,187\,8128\times10^{-12}$ F m^{-1} es la permitividad eléctrica del vacío, y $\mathbf{r}_i$ es el vector posición del electrón i en el sistema de referencia elegido. El potencial debido a la atracción entre el protón I y el electrón i está dado por:

$$V_{iI} = -\frac{1}{4\pi\varepsilon_0}\frac{eZ_I}{|\mathbf{r}_i - \mathbf{R}_I|} \tag{16}$$

donde Z_I es el número atómico del núcleo I (por ejemplo, en un átomo de helio $Z = 2$) y $\mathbf{R}_I$ es la posición de dicho núcleo en el sistema de referencia elegido.

El sistema de referencia usualmente se toma en el **centro de masa** o en el **centro de la molécula**. El potencial dado por la repulsión entre los núcleos I y J (protones) es:

$$\hat{H} = -\sum_{i=1}^{N}\frac{1}{2}\nabla_i^2 - \sum_{I=1}^{M}\frac{1}{2M_I}\nabla_I^2 - \sum_{i=1}^{N}\sum_{I=1}^{M}\frac{Z_I}{|\mathbf{r}_i - \mathbf{R}_I|} + \sum_{1\le i<j\le N}\frac{1}{|\mathbf{r}_i - \mathbf{r}_j|} + \sum_{1\le I<J\le M}\frac{Z_I Z_J}{|\mathbf{R}_I - \mathbf{R}_J|} \tag{17}$$

4.0.3 Condiciones de la solución

Como en cualquier ecuación diferencial, cuando se busca una solución es importante considerar las condiciones iniciales (CI) y las condiciones de frontera (CF); aquí debemos satisfacer ciertas condiciones que provienen de leyes físicas.

A nivel microscópico, las partículas son indistinguibles: ninguna medición puede distinguir entre el electrón uno y el electrón dos. Por lo tanto, intercambiar las etiquetas de dos partículas debe dejar inalteradas todas las predicciones medibles.

La teoría cuántica [17] muestra que las partículas son bosones (p. ej., fotones), que siguen la estadística de **Bose–Einstein** [18, 19], o fermiones (p. ej., electrones, protones), que siguen la estadística de **Fermi–Dirac** [20, 21]. Una consecuencia importante de esta última es la antisimetría de la función de onda.

Si sustituimos las mismas coordenadas $x_1 = x_2$ (lo que significa la misma posición y la misma coordenada de espín),

$$\psi(x, x) = -\psi(x, x) \implies \psi(x, x) = 0 \tag{18}$$

entonces dos fermiones no pueden ocupar el mismo estado cuántico de una partícula. Esto es el principio de exclusión de Pauli [22]. De manera más general:

$$\psi(\dots, \mathbf{x}_i, \dots, \mathbf{x}_j, \dots) = -\psi(\dots, \mathbf{x}_j, \dots, \mathbf{x}_i, \dots) \tag{19}$$

Podemos imponer la **antisimetría** usando un determinante de Slater $N \times N$ [23], que involucra únicamente estados de una partícula (es decir, una función de onda con una sola entrada). Un intercambio de cualquier par de partículas corresponde a un intercambio de dos columnas del determinante; dicho intercambio introduce un cambio de signo en el determinante. Para permutaciones pares se tiene $(-1)^P = 1$, y para permutaciones impares se tiene $(-1)^P = -1$.

$$\psi(\mathbf{x}_1, \dots, \mathbf{x}_N) \propto \begin{vmatrix} \phi_1(\mathbf{x}_1) & \phi_2(\mathbf{x}_1) & \cdots & \phi_N(\mathbf{x}_1) \\ \phi_1(\mathbf{x}_2) & \phi_2(\mathbf{x}_2) & \cdots & \phi_N(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_1(\mathbf{x}_N) & \phi_2(\mathbf{x}_N) & \cdots & \phi_N(\mathbf{x}_N) \end{vmatrix} \tag{20}$$

Done ϕ_k es una funcion de onda de un unica entrada llamada spin orbital. Cuando considerando dos electrones: $(N = 2)$:

$$\Psi(\mathbf{x}_1, \mathbf{x}_2) \propto \Big[\phi_a(\mathbf{x}_1)\phi_b(\mathbf{x}_2) - \phi_a(\mathbf{x}_2)\phi_b(\mathbf{x}_1)\Big]. \tag{21}$$

La energía potencial se vuelve infinita cuando dos partículas se superponen, lo cual impone restricciones estrictas sobre la forma de la función de onda en esos puntos. Estas restricciones se conocen como las **condiciones de cúspide de Kato** [24]. Las condiciones de cúspide establecen que la función de onda debe ser no diferenciable en esos puntos y proporcionan valores exactos para las derivadas promedio en las cúspides. Más precisamente, los resultados de cúspide del teorema de Kato son:

Cúspide electrón–núcleo (electrón con carga -1, núcleo con carga $+Z$, masa reducida $\mu \approx 1$)

$$\lim_{r_{iI} \to 0} \left(\frac{\partial \psi}{\partial r_{iI}} \right) = -Z\psi(r_{iI} = 0) \tag{22}$$

Cúspide electrón–electrón, espines opuestos (cargas -1, masa reducida $\mu = \frac{1}{2}$)

$$\lim_{r_{ij} \to 0} \left(\frac{\partial \psi}{\partial r_{ij}} \right) = \frac{1}{2}\psi(r_{ij} = 0) \tag{23}$$

donde r_{iI} (r_{ij}) es la distancia electrón–núcleo (electrón–electrón), Z_I es la carga nuclear del I-ésimo núcleo, y "ave" implica un promedio esférico sobre todas las direcciones.

Estas condiciones pueden obtenerse si multiplicamos el ansatz por un factor de Jastrow $\mathcal{J}$, el cual satisface estas condiciones de manera analítica [25].

Así, el problema consiste en encontrar una ψ que satisfaga todas esas condiciones.

4.0.4 Aproximaciones al problema

Es claro que encontrar soluciones analíticas es prácticamente imposible, por lo que lo que se ha hecho es aplicar primero buenas aproximaciones. La **aproximación de Born–Oppenheimer** [3] permite separar el movimiento de los núcleos y el movimiento de los electrones; al describir los electrones en una molécula, se desprecia el movimiento de los núcleos atómicos. La base física de la aproximación de Born–Oppenheimer es que la masa de un núcleo atómico en una molécula es mucho mayor que la masa de un electrón (más de 1000 veces) [26]. Debido a esta diferencia, los núcleos se mueven mucho más lentamente que los electrones. Además, debido a sus cargas opuestas, existe una fuerza atractiva mutua que actúa sobre un núcleo atómico y un electrón. Esta fuerza provoca que ambas partículas se aceleren. Como la magnitud de la aceleración es inversamente proporcional a la masa, $a = f/m$ [27], la aceleración de los electrones es grande y la aceleración de los núcleos atómicos es pequeña; la diferencia es un factor mayor que 1000. En consecuencia, los electrones se mueven y responden a las fuerzas muy rápidamente, mientras que los núcleos no. Entonces fijamos los núcleos: $\mathbf{R}_I$ pasa a ser una constante, por lo que el término cinético de los núcleos se vuelve cero. Además, la energía potencial debida a la repulsión entre núcleos se vuelve una constante.

$$\begin{aligned} \hat{H}_{\text{el}} = &- \sum_{i=1}^{N} \frac{1}{2}\nabla_i^2 - \sum_{i=1}^{N}\sum_{I=1}^{M} \frac{Z_I}{|\mathbf{r}_i - \mathbf{R}_I|} \\ &+ \sum_{1 \le i < j \le N} \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|} + \sum_{I<J} \frac{Z_I Z_J}{|\mathbf{R}_I - \mathbf{R}_J|} \end{aligned} \tag{24}$$

Esta es la forma con la que vamos a trabajar; un segundo enfoque útil para el problema es usar un **ansatz**, que es una solución tentativa guiada por la intuición. Normalmente depende de cierto número de parámetros; entonces el problema se convierte en optimizar este **ansatz**.

4.0.5 Cociente de Rayleigh

Necesitamos una forma de medir qué tan bien se comporta nuestro ansatz ψ_θ. Si nuestro ansatz es "malo", entonces no refleja la forma real del sistema. La mayoría de enfoques de aprendizaje profundo usan conjuntos de datos para entrenar su ansatz (modelo inicializado con parámetros aleatorios); en este caso no necesitamos nada externo, sino únicamente principios físicos. Introduzcamos primero el **cociente de Rayleigh**. Si A es un operador y x es un estado, el número:

$$R_A(x) = \frac{\langle x|A|x\rangle}{\langle x|\,x\rangle} \tag{25}$$

es el **valor de expectativa** de ese operador en ese estado. Lo importante para nosotros es que si ψ es una función de onda y $\hat{H}$ es el **Hamiltoniano**, entonces el

cociente de Rayleigh:

$$R_{\hat{H}}(\psi) = \frac{\langle\psi|\hat{H}|\psi\rangle}{\langle\psi|\psi\rangle} \tag{26}$$

es la energía promedio (esperada) del sistema cuando se encuentra en el estado ψ [28].

4.0.6 Principio variacional

El principio variacional para sistemas electrónicos establece que el valor de expectativa de la energía de enlace obtenido usando una función de onda aproximada y el operador Hamiltoniano exacto será mayor o igual que la energía verdadera del sistema [29]. Esta idea es realmente poderosa. Cuando se implementa, nos permite encontrar la mejor función de onda aproximada dentro de una familia de funciones de onda que contienen uno o más parámetros ajustables, llamada función de onda de prueba [30]. Un enunciado matemático del principio variacional es:

$$R_{\hat{H}}(\psi_{\text{ansatz}}) \geq R_{\hat{H}}(\psi_{\text{true}}) \tag{27}$$

La verdadera función de onda del estado fundamental ψ_{true} es aquella que minimiza el cociente de Rayleigh:

$$\psi_0 = \underset{\psi}{\operatorname{argmin}}(R_{\hat{H}}(\psi)) \tag{28}$$

Así, si minimizamos el cociente de Rayleigh de nuestro ansatz, nos vamos a acercar más a la función de onda verdadera.

4.0.7 Monte Carlo variacional

El proceso que vamos a utilizar para optimizar nuestro ansatz se llama Monte Carlo Variacional (VMC) [4, 31]. Ahora podemos ver el cociente de Rayleigh como una función de pérdida $\mathcal{L}$ de la forma:

$$\mathcal{L}(\Psi_\theta) = \frac{\langle\Psi_\theta|\,\hat{H}\,|\Psi_\theta\rangle}{\langle\Psi_\theta|\Psi_\theta\rangle} = \frac{\int d\mathbf{r}\Psi^*(\mathbf{r})\hat{H}\Psi(\mathbf{r})}{\int d\mathbf{r}\Psi^*(\mathbf{r})\Psi(\mathbf{r})} \tag{29}$$

Evaluar esa integral es difícil; otro enfoque inteligente es el siguiente: definimos una distribución de probabilidad p_θ con la forma:

$$p_\theta(x) = \frac{|\Psi_\theta(x)|^2}{\int dx'\Psi_\theta^2(x')} \tag{30}$$

Nótese que calcular $p_\theta(x)$ para un x específico es complicado debido a la integral que aparece; esto será importante más adelante. Definimos la energía local E_L como:

$$E_L(x) = \Psi_\theta^{-1}(x)\hat{H}\Psi_\theta(x) \tag{31}$$

Entonces la pérdida se convierte en:

$$\mathcal{L}(\Psi_\theta) = \int \frac{\hat{H}\Psi_\theta(x)}{\Psi_\theta(x)} p_\theta(x)dx \tag{32}$$

la cual es el valor esperado de la energía local:

$$\mathcal{L}(\Psi_\theta) = \mathbb{E}_{x\sim p_\theta}[E_L(x)] \tag{33}$$

Para optimizar nuestra función de onda necesitamos calcular este valor esperado y obtener su derivada para la retropropagación. Usaremos el estimador de Monte Carlo [32]:

$$\mathcal{L}_\theta = \mathbb{E}_{x\sim p_\theta}[E_L(x)] \approx \frac{1}{M}\sum_{i=1}^{M} E_L(\mathbf{R}_k) \tag{34}$$

donde $\mathbf{R}_k$ son muestras de la distribución de probabilidad p_θ, es decir:

$$\mathbf{R}_1, \ldots, \mathbf{R}_M \sim p_\theta(\mathbf{R})$$

Obtenemos $E_L(\mathbf{R}_k)$ usando:

$$E_L(\mathbf{R}_k) = \frac{\hat{H}\psi(\mathbf{R}_k)}{\psi(\mathbf{R}_k)} = -\frac{1}{2}\frac{\nabla^2\psi(\mathbf{R_k})}{\psi(\mathbf{R}_k)} + V(\mathbf{R}_k) \tag{35}$$

Dado que, en la práctica, es mejor trabajar en el espacio logarítmico por ser numéricamente más estable, usamos la siguiente forma para el término cinético. Para cualquier función derivable f sabemos que:

$$\frac{\nabla^2 f}{f} = \left[\nabla^2 \log f + (\nabla \log f)^2\right] \tag{36}$$

Por lo tanto, obtenemos:

$$E_L(\mathbf{R}_k) = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{3}\left[\frac{\partial^2 \log|\Psi(x)|}{\partial r_{ij}^2} + \left(\frac{\partial \log|\Psi(x)|}{\partial r_{ij}}\right)^2\right] + V(\mathbf{R_k}) \tag{37}$$

El gradiente de la energía respecto a los parámetros de una función de onda parametrizada es:

$$\nabla_\theta\mathcal{L} = 2\mathbb{E}_{x\sim\Psi^2}[(E_L(x) - \mathbb{E}_{x'\sim\Psi^2}[E_L(x')])\nabla \log|\Psi(x)|] \tag{38}$$

4.0.8 Algoritmo de Metropolis–Hastings

Para obtener muestras de la distribución de probabilidad p_θ vamos a usar el algoritmo de Metropolis–Hastings (MH) [33], que es un método de Monte Carlo por cadenas de Markov (MCMC) utilizado para obtener una secuencia de muestras aleatorias de una distribución de probabilidad. La razón para usar este método en lugar de otros métodos bien conocidos (p. ej., *example*) es que MH no sufre de la *maldición de la dimensionalidad* [34]; es decir, el costo computacional no explota al aumentar la dimensión del problema, y dado que trabajaremos en altas dimensiones, es una buena opción utilizar este método. El algoritmo funciona de la siguiente manera:

1. Tomar una configuración inicial $\mathbf{X}_0 \in E$ arbitraria.
2. Proponer $\mathbf{X}' = \mathbf{X}_0 + \eta$, donde $\eta \sim q(\eta)$. Aquí q es una densidad de probabilidad sobre E llamada **núcleo de propuesta**. En nuestro caso, muestreamos de una gaussiana simétrica.

3. Calcular la cantidad:

$$A(\mathbf{X_0}, \mathbf{X}') = \min\left(1, \frac{\rho(\mathbf{X}')}{\rho(\mathbf{X}_0)} \frac{q(\mathbf{X}' - \mathbf{X}_0)}{q(\mathbf{X}_0 - \mathbf{X}')}\right)$$

donde ρ es la distribución objetivo (la que queremos muestrear). En el caso en que q, el núcleo de propuesta, es simétrico, esto se simplifica a:

$$A(\mathbf{X}_0, \mathbf{X}') = \min\left(1, \frac{\rho(\mathbf{X}')}{\rho(\mathbf{X}_0)}\right) \tag{39}$$

Nótese que, en nuestro caso, ρ es igual a p_θ. Dijimos que calcular el factor integral es un desafío, pero aquí no importa porque:

$$\frac{p_\theta(\mathbf{X}')}{p_\theta(\mathbf{X}_0)} = \frac{|\psi_\theta(\mathbf{X}')|^2 / \int |\psi_\theta|^2 dx}{|\psi_\theta(\mathbf{X}_0)|^2 / \int |\psi_\theta|^2 dx} = \frac{|\psi_\theta(\mathbf{X}')|^2}{|\psi_\theta(\mathbf{X}_0)|^2} \tag{40}$$

4. Generar un número uniforme $U \in [0, 1]$.

5. Si $U < A(\mathbf{X}_0 \to \mathbf{X}'_l)$, entonces $\mathbf{X_1} = \mathbf{X}'$; en caso contrario, se intenta con otra $\mathbf{X}'$.

6. Repetir hasta obtener N_{eq} muestras aceptadas; para un N_{eq} grande los cambios entre muestras se estabilizan (se alcanza una distribución estacionaria). Esta fase se denomina **burn-in**.

7. A partir de $\mathbf{X}_{N_{\text{eq}}}$ generar otras M muestras hasta obtener la muestra $\mathbf{X}_{N_{\text{eq}}+M+1}$. En cada muestra se calcula $E_L(\mathbf{R}_k)$ y luego se promedia para obtener $\mathbb{E}(E_L)$.

Una vez que se obtiene $\mathbb{E}(E_L)$ y usando la ecuación 38, podemos optimizar nuestra red usando retropropagación [35].

4.1 Fundamentos de aprendizaje profundo

Esta subsección introduce los conceptos centrales de aprendizaje profundo que se aplicarán en este trabajo.

4.1.1 Perceptrón multicapa

Un perceptrón multicapa (MLP) es un mapeo no lineal $\mathcal{F} : \mathbb{R}^{\text{in}} \to \mathbb{R}^{\text{out}}$ desde un patrón de entrada $\mathbf{x}$ hacia un vector de salida $\mathbf{y}$ [35], y es la composición de L capas: la primera se llama capa de entrada, la última, capa de salida, y las intermedias, capas ocultas. En cada capa encontramos un número arbitrario de neuronas y una transformación afín $\mathbf{z}^{(l)}$, $l \in \{L, L-1, \ldots, 2\}$ (con $l = 1$ como la capa de entrada) de la forma siguiente.

$$\mathbf{z}^{(l)} = \mathbf{W}^{(l)} \mathbf{a}^{(l-1)} + \mathbf{b}^{(l)} \tag{41}$$

donde $\mathbf{W}^{(l)}$ se denomina la matriz de pesos y $\mathbf{b}^{(l)}$ es el vector de sesgo de la capa l. Usamos una función no lineal $\sigma^{(l)}$ en la capa l (típicamente Softmax, ReLU, Tanh); por lo tanto, la salida de cada capa es:

$$f^{(l)} = \sigma^{(l)} \circ \mathbf{z}^{(l)} \tag{42}$$

donde $\circ$ denota la composición de funciones. Un MLP es la composición de todas las capas:

$$\mathcal{F} = f^{(L)} \circ f^{(L-1)} \circ \cdots \circ f^{(1)} \tag{43}$$

Llamamos **parámetros** al conjunto de todos los pesos y sesgos de cada capa; lo representamos con el símbolo θ:

$$\{\mathbf{W}^{(l)}, \mathbf{b}^{(l)}\}_{l=2}^{L} = \theta$$

Típicamente se entrena un MLP usando un conjunto de datos de entrenamiento, una función de pérdida (p. ej., error cuadrático medio, error absoluto medio, entropía cruzada) y un optimizador (p. ej., GD, SGD, ADAM). Adicionalmente, se pueden usar técnicas de regularización como *dropout* para mejorar la capacidad de generalización de la red [36, 37, 38, 39].

4.1.2 Descenso por gradiente natural

Como se mencionó anteriormente, existen muchas maneras de actualizar los parámetros. Todas ellas asumen implícitamente que el espacio de parámetros $\Theta \subset \mathbb{R}^d$ está equipado con la métrica euclidiana estándar, de modo que la "longitud" y el "descenso más pronunciado" se miden con respecto a $\|\Delta\theta\|_2$.

En nuestro caso, la pérdida $\mathcal{L}(\theta)$ depende de una distribución de probabilidad p_θ, no solo de θ de manera directa. Por ejemplo, en Monte Carlo variacional trabajamos con la ecuación eq. (30), donde θ parametriza toda una familia de densidades de probabilidad sobre configuraciones x. Por ello, es más natural medir distancias entre *distribuciones* p_θ y $p_{\theta+\Delta\theta}$, en lugar de hacerlo entre los vectores de parámetros en sí.

Una forma canónica de medir la distancia entre distribuciones de probabilidad cercanas es la divergencia de Kullback–Leibler (KL) [40].

$$\mathrm{KL}\big(p_\theta \,\|\, p_{\theta+\Delta\theta}\big) = \mathbb{E}_{x\sim p_\theta}\left[\log \frac{p_\theta(x)}{p_{\theta+\Delta\theta}(x)}\right] \tag{44}$$

Para pasos pequeños $\Delta\theta$, puede mostrarse que una expansión de Taylor de segundo orden de la KL produce

$$\mathrm{KL}\big(p_\theta \,\|\, p_{\theta+\Delta\theta}\big) = \tfrac{1}{2}\, \Delta\theta^\top \mathcal{F}(\theta)\, \Delta\theta + \mathcal{O}(\|\Delta\theta\|^3),$$

donde $\mathcal{F}(\theta)$ es la Matriz de Información de Fisher (FIM) [41]. Para definirla, introducimos la **función score**

$$s_\theta(x) = \nabla_\theta \log p(x \mid \theta) \in \mathbb{R}^d,$$

entonces la FIM es:

$$\mathcal{F}(\theta) = \mathbb{E}_{x\sim p(\cdot|\theta)}\big[\, s_\theta(x)\, s_\theta(x)^{\mathsf{T}}\big]. \tag{45}$$

El conjunto de distribuciones:

$$\mathcal{M} = \{\, p_\theta(z) \mid \theta \in \Theta \subset \mathbb{R}^d \,\} \tag{46}$$

puede verse como una variedad diferenciable, y $\mathcal{F}(\theta)$ define una métrica riemanniana sobre su espacio tangente. Concretamente, para vectores tangentes $u, v \in \mathbb{R}^d$ en θ definimos el producto interno:

$$\langle u, v\rangle_\theta = u^{\mathsf{T}}\, \mathcal{F}(\theta)\, v. \tag{47}$$

Esta métrica dice: dos direcciones de parámetros son "cercanas" si inducen cambios infinitesimales similares en la *distribución* p_θ.

Para obtener la dirección de descenso más pronunciado con esta métrica no euclidiana, es necesario resolver un problema de optimización con restricción: encontrar una variación $\Delta\theta$ que disminuya $\mathcal{L}(\theta)$ lo más rápido posible, entre todas las direcciones con "longitud" fija $\|\Delta\theta\|_\theta^2 = \Delta\theta^\top \mathcal{F}(\theta)\,\Delta\theta$. Esa dirección se llama la dirección de **gradiente natural** y toma la forma [42]:

$$\Delta\theta_{\text{nat}} \;\propto\; -\,\mathcal{F}(\theta)^{-1}\,\nabla_\theta \mathcal{L}(\theta). \tag{48}$$

Así, la actualización por descenso de gradiente natural es:

$$\Delta\theta_{\text{nat}} = -\,\eta\,\mathcal{F}(\theta)^{-1}\,\nabla_\theta \mathcal{L}(\theta) \tag{49}$$

donde $\eta > 0$ es un tamaño de paso. Comparado con el gradiente usual $\nabla_\theta \mathcal{L}$, el factor $\mathcal{F}^{-1}$ "precondiciona" el gradiente mediante la geometría local de la distribución de probabilidad del modelo: direcciones que apenas cambian p_θ se amplifican, y direcciones que lo cambian mucho se atenúan. El descenso por gradiente natural es, por lo tanto, significativo justamente en la situación que nos interesa: cuando la pérdida depende de los parámetros *a través* de un modelo probabilístico p_θ (p. ej., verosimilitud, entropía cruzada, KL, objetivos variacionales, energía de Monte Carlo variacional, etc.) [42].

4.1.3 Curvatura aproximada factorizada de Kronecker

Calcular e invertir directamente la matriz de Fisher completa $\mathcal{F}(\theta)$ es inviable para redes neuronales modernas, ya que θ puede tener millones de componentes. La Curvatura Aproximada Factorizada de Kronecker (KFAC) [43] es una aproximación eficiente que hace prácticas las actualizaciones de gradiente natural en redes por capas. Esbozamos la construcción para una capa totalmente conectada ℓ con matriz de pesos W_ℓ y (por simplicidad) sin sesgo. Los términos de sesgo pueden incluirse aumentando las activaciones con una constante 1; comentamos esto a continuación.

Comenzamos con las definiciones hacia adelante. Sea $\mathbf{a}_\ell$ la **entrada (activación) de la capa** ℓ. Este es el vector columna de activaciones que llegan a la capa ℓ. Para la primera capa oculta, $\mathbf{a}_1$ es simplemente la entrada (posiblemente preprocesada). Para capas más profundas, corresponde a la salida de la no linealidad de la capa anterior.

La **pre-activación en la capa** ℓ es simplemente la cantidad:

$$\mathbf{h}_\ell = W_\ell\,\mathbf{a}_\ell \tag{50}$$

y la **activación de salida de la capa** ℓ es:

$$\tilde{\mathbf{a}}_\ell = \phi(\mathbf{h}_\ell) \tag{51}$$

donde ϕ se aplica elemento a elemento. En muchas notaciones, $\tilde{\mathbf{a}}_\ell$ pasaría a ser la entrada de la siguiente capa, pero aquí mantenemos esta notación para que sea consistente con el bloque de Fisher asociado a W_ℓ.

Ahora pasamos a las definiciones hacia atrás. Sea la pérdida para una sola muestra $\mathcal{L}(\theta)$ (por ejemplo, el logaritmo negativo de la verosimilitud, o el logaritmo negativo de la probabilidad asociada a la función de onda). Definimos la **sensibilidad hacia atrás** (o señal de error) en la capa ℓ como:

$$\mathbf{e}_\ell = \frac{\partial \mathcal{L}}{\partial \mathbf{h}_\ell} \in \mathbb{R}^{m_\ell} \tag{52}$$

Esto se calcula mediante retropropagación. En la capa de salida L:

$$\mathbf{e}_L = \frac{\partial \mathcal{L}}{\partial \mathbf{h}_L} = \left(\frac{\partial \mathcal{L}}{\partial \tilde{\mathbf{a}}_L}\right) \odot \phi'(\mathbf{h}_L) \tag{53}$$

donde $\odot$ es el producto elemento a elemento, también llamado producto de Hadamard. Para capas ocultas $\ell < L$:

$$\mathbf{e}_\ell = \frac{\partial \mathcal{L}}{\partial \mathbf{h}_\ell} = \left(W_{\ell+1}^\mathsf{T}\mathbf{e}_{\ell+1}\right) \odot \phi'(\mathbf{h}_\ell) \tag{54}$$

En el contexto del gradiente natural para modelos probabilísticos, $\mathcal{L}$ suele elegirse como $-\log p(X \mid \theta)$, así que, salvo por un signo, también podemos pensar en $\mathbf{e}_\ell$ como:

$$\mathbf{e}_\ell = \frac{\partial \log p(X \mid \theta)}{\partial \mathbf{h}_\ell}.$$

Para una sola muestra, usando la regla de la cadena,

$$\frac{\partial \mathcal{L}}{\partial W_\ell} = \frac{\partial \mathcal{L}}{\partial \mathbf{h}_\ell}\frac{\partial \mathbf{h}_\ell}{\partial W_\ell} = \mathbf{e}_\ell\,\mathbf{a}_\ell^\mathsf{T} \tag{55}$$

Si en lugar de $\mathcal{L}$ usamos $\log p(X \mid \theta)$ (como en la definición de Fisher), obtenemos:

$$\frac{\partial \log p(X \mid \theta)}{\partial W_\ell} = \mathbf{e}_\ell\,\mathbf{a}_\ell^\mathsf{T},$$

con $\mathbf{e}_\ell = \partial \log p / \partial \mathbf{h}_\ell$. Ahora vectorizamos el gradiente. Usando la identidad estándar:

$$\operatorname{vec}(uv^\mathsf{T}) = v \otimes u,$$

with $u = \mathbf{e}_\ell$ and $v = \mathbf{a}_\ell$, we obtain

$$\frac{\partial \log p(X \mid \theta)}{\partial \operatorname{vec}(W_\ell)} = \operatorname{vec}\left(\frac{\partial \log p}{\partial W_\ell}\right) = \operatorname{vec}(\mathbf{e}_\ell\,\mathbf{a}_\ell^\mathsf{T}) = \mathbf{a}_\ell \otimes \mathbf{e}_\ell.$$

Esto da la forma estructural clave utilizada por KFAC. El bloque de Fisher asociado a los parámetros W_ℓ es:

$$\mathcal{F}_\ell = \mathbb{E}_{p(\mathbf{X})}\left[\frac{\partial \log p(X \mid \theta)}{\partial \operatorname{vec}(W_\ell)}\frac{\partial \log p(X \mid \theta)}{\partial \operatorname{vec}(W_\ell)}^\mathsf{T}\right].$$

Sustituyendo la expresión anterior,

$$\mathcal{F}_\ell = \mathbb{E}_{p(\mathbf{X})}\big[(\mathbf{a}_\ell \otimes \mathbf{e}_\ell)(\mathbf{a}_\ell \otimes \mathbf{e}_\ell)^\mathsf{T}\big]. \tag{56}$$

Aquí $p(\mathbf{X})$ denota la distribución sobre entradas y etiquetas (o configuraciones, en el caso de VMC). En la práctica, este valor esperado se aproxima promediando sobre un mini-lote de muestras X y las correspondientes pasadas hacia adelante/atrás que producen $\mathbf{a}_\ell$ y $\mathbf{e}_\ell$.

Calcular e invertir $\mathcal{F}_\ell$ directamente sigue siendo costoso, porque su dimensión es:

$$(\dim(\mathbf{a}_\ell)\dim(\mathbf{e}_\ell)) \times (\dim(\mathbf{a}_\ell)\dim(\mathbf{e}_\ell)).$$

KFAC introduce dos aproximaciones clave para que esto sea tratable.

1. **Bloque diagonal por capas.**
   Se asume que los bloques fuera de la diagonal $\mathcal{F}_{ij}$ son despreciables cuando θ_i y θ_j pertenecen a capas diferentes. Esto hace que la matriz de Fisher sea aproximadamente bloque-diagonal, con un bloque por capa.

2. **Factorización de Kronecker dentro de cada capa.**
   Dentro de una capa, KFAC supone que la correlación entre activaciones y errores se factoriza:

$$\begin{aligned}\mathcal{F}_\ell =& \mathbb{E}_{p(\mathbf{X})}\big[(\mathbf{a}_\ell \otimes \mathbf{e}_\ell)(\mathbf{a}_\ell \otimes \mathbf{e}_\ell)^\mathsf{T}\big] \\ =& \mathbb{E}_{p(\mathbf{X})}\big[(\mathbf{a}_\ell\mathbf{a}_\ell^\mathsf{T}) \otimes (\mathbf{e}_\ell\mathbf{e}_\ell^\mathsf{T})\big] \\ \approx& \mathbb{E}_{p(\mathbf{X})}[\mathbf{a}_\ell\mathbf{a}_\ell^\mathsf{T}] \otimes \mathbb{E}_{p(\mathbf{X})}[\mathbf{e}_\ell\mathbf{e}_\ell^\mathsf{T}] \end{aligned} \tag{57}$$

Definimos la *covarianza de activaciones* y la *covarianza del error*:

$$A_\ell = \mathbb{E}_{p(\mathbf{X})}[\mathbf{a}_\ell\mathbf{a}_\ell^\mathsf{T}], \qquad S_\ell = \mathbb{E}_{p(\mathbf{X})}[\mathbf{e}_\ell\mathbf{e}_\ell^\mathsf{T}].$$

En la práctica, estos valores esperados se actualizan como promedios móviles sobre mini-lotes:

$$A_\ell \approx \frac{1}{B}\sum_{b=1}^{B} \mathbf{a}_\ell^{(b)}\mathbf{a}_\ell^{(b)\mathsf{T}}, \qquad S_\ell \approx \frac{1}{B}\sum_{b=1}^{B} \mathbf{e}_\ell^{(b)}\mathbf{e}_\ell^{(b)\mathsf{T}},$$

donde b indexa las muestras en el lote y $\mathbf{a}_\ell^{(b)}, \mathbf{e}_\ell^{(b)}$ se obtienen mediante una pasada estándar hacia adelante y hacia atrás para esa muestra. Con esta aproximación tenemos:

$$\mathcal{F}_\ell \approx A_\ell \otimes S_\ell.$$

La propiedad crucial del producto de Kronecker es que:

$$(A_\ell \otimes S_\ell)^{-1} = A_\ell^{-1} \otimes S_\ell^{-1},$$

por lo que la inversa del (enorme) bloque de Fisher por capa puede obtenerse invirtiendo las matrices mucho más pequeñas A_ℓ y S_ℓ. Así, la actualización de gradiente natural para los pesos de la capa ℓ queda:

$$\Delta\theta_{\mathrm{nat},\ell} \approx -\,\eta\,\big(A_\ell^{-1} \otimes S_\ell^{-1}\big)\,\nabla_{\mathrm{vec}(W_\ell)}\mathcal{L}. \tag{58}$$

En resumen, KFAC reemplaza la inversa intratable

$$\mathbb{E}_{p(\mathbf{X})}\big[(\mathbf{a}_\ell \otimes \mathbf{e}_\ell)(\mathbf{a}_\ell \otimes \mathbf{e}_\ell)^\mathsf{T}\big]^{-1}$$

por la siguiente expresión, que puede calcularse de manera eficiente:

$$\begin{aligned}\mathcal{F}_\ell^{-1} &= \mathbb{E}_{p(\mathbf{X})}\big[(\mathbf{a}_\ell \otimes \mathbf{e}_\ell)(\mathbf{a}_\ell \otimes \mathbf{e}_\ell)^\mathsf{T}\big]^{-1} \\ &\approx \mathbb{E}_{p(\mathbf{X})}[\mathbf{a}_\ell\mathbf{a}_\ell^\mathsf{T}]^{-1} \otimes \mathbb{E}_{p(\mathbf{X})}[\mathbf{e}_\ell\mathbf{e}_\ell^\mathsf{T}]^{-1} \end{aligned} \tag{59}$$

lo cual captura la estructura dominante de la curvatura mientras mantiene el costo del descenso por gradiente natural comparable al de los métodos estándar de primer orden. Hemos ignorado los sesgos arriba por claridad. En la práctica, se puede (i) aumentar $\mathbf{a}_\ell$ con una constante 1 para absorber los sesgos en W_ℓ, o (ii) mantener factores KFAC separados y más pequeños para los sesgos; ambos enfoques preservan la misma estructura de Kronecker.

4.1.4 Autoatención y autoatención multi-cabeza

La idea de un *mecanismo de atención* fue introducida en traducción automática neuronal por Bahdanau et al. [44]. En lugar de comprimir toda una secuencia de entrada en un único vector de tamaño fijo, el modelo aprende a **enfocarse** en diferentes partes de la entrada al generar cada token de salida.

Dado un vector *query* (consulta) $\mathbf{q} \in \mathbb{R}^{d_h}$ y un conjunto de pares clave–valor $\{(\mathbf{k}_j, \mathbf{v}_j)\}_{j=1}^{T}$ con $\mathbf{k}_j, \mathbf{v}_j \in \mathbb{R}^{d_h}$, el mecanismo de atención (producto punto escalado) calcula:

1. **Puntajes de compatibilidad**

$$e_j \;=\; \frac{\mathbf{q}^\top \mathbf{k}_j}{\sqrt{d_h}}, \qquad j = 1, \ldots, T,$$

2. **Pesos de atención normalizados**

$$\alpha_j \;=\; \frac{\exp(e_j)}{\sum_{m=1}^{T}\exp(e_m)} \;=\; \mathrm{Softmax}_j\!\left(\frac{\mathbf{q}^\top \mathbf{k}_j}{\sqrt{d_h}}\right),$$

3. **Suma ponderada de valores**

$$\mathbf{o} \;=\; \sum_{j=1}^{T} \alpha_j \mathbf{v}_j.$$

Intuitivamente, la consulta $\mathbf{q}$ pregunta: *"¿qué elementos del conjunto son relevantes para mí ahora?"* Las claves $\mathbf{k}_j$ codifican *lo que ofrece cada elemento*, y los valores $\mathbf{v}_j$ codifican *lo que extraemos de cada elemento una vez que decidimos prestarle atención.*

En **autoatención**, las consultas, claves y valores se obtienen del **mismo** conjunto de vectores de entrada. Consideremos una secuencia de *embeddings* de entrada

$$\mathbf{x}_1, \ldots, \mathbf{x}_T \in \mathbb{R}^d,$$

y apilémoslos en una matriz

$$\mathbf{X} \in \mathbb{R}^{T\times d}, \quad \mathbf{X} = \begin{bmatrix} \mathbf{x}_1^\top \\ \vdots \\ \mathbf{x}_T^\top \end{bmatrix}.$$

Para construir una **cabeza** de atención de dimensión d_h, introducimos tres matrices entrenables:

$$\mathbf{W}^Q \in \mathbb{R}^{d\times d_h}, \quad \mathbf{W}^K \in \mathbb{R}^{d\times d_h}, \quad \mathbf{W}^V \in \mathbb{R}^{d\times d_h}.$$

Luego calculamos consultas, claves y valores:

$$\mathbf{Q} = \mathbf{X}\mathbf{W}^Q \in \mathbb{R}^{T\times d_h}$$
$$\mathbf{K} = \mathbf{X}\mathbf{W}^K \in \mathbb{R}^{T\times d_h}$$
$$\mathbf{V} = \mathbf{X}\mathbf{W}^V \in \mathbb{R}^{T\times d_h}$$

La **autoatención (producto punto escalado)** para esta cabeza es:

$$\text{Attention}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \text{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_h}}\right)\mathbf{V} \qquad (60)$$

donde la softmax se aplica por filas. Elemento a elemento, la salida en la posición t es:

$$\mathbf{o}_t = \sum_{j=1}^{T} \alpha_{tj}\mathbf{v}_j \text{ con } \alpha_{tj} = \frac{\exp\left(\mathbf{q}_t^\top \mathbf{k}_j/\sqrt{d_h}\right)}{\sum_{m=1}^{T}\exp\left(\mathbf{q}_t^\top \mathbf{k}_m/\sqrt{d_h}\right)}.$$

Puede pensarse como: *cada posición t en la secuencia "mira" a todas las demás posiciones j y decide cuánto le importa cada una.*

Una sola cabeza solo puede capturar interacciones en un único "subespacio de representación" de dimensión d_h. La **atención multi-cabeza** usa varias cabezas en paralelo, cada una con sus propias matrices de proyección, de modo que se puedan capturar simultáneamente diferentes tipos de relaciones. Sea n_h el número de cabezas. Para la cabeza $i = 1,\ldots,n_h$ tenemos:

$$\mathbf{W}_i^Q,\ \mathbf{W}_i^K,\ \mathbf{W}_i^V \in \mathbb{R}^{d\times d_h}.$$

La cabeza i calcula:

$$\text{head}_i(\mathbf{X}) = \text{Attention}(\mathbf{X}\mathbf{W}_i^Q,\ \mathbf{X}\mathbf{W}_i^K,\ \mathbf{X}\mathbf{W}_i^V) \in \mathbb{R}^{T\times d_h}.$$

Las salidas de todas las cabezas se concatenan a lo largo de la dimensión de características y luego se mezclan linealmente:

$$\mathbf{U} = [\text{head}_1(\mathbf{X})\,;\,\ldots\,;\,\text{head}_{n_h}(\mathbf{X})] \in \mathbb{R}^{T\times(n_h d_h)},$$

$$\mathbf{O} = \mathbf{U}\mathbf{W}^O, \qquad \mathbf{W}^O \in \mathbb{R}^{(n_h d_h)\times d}.$$

Si nos enfocamos en un paso temporal t y en la cabeza i, podemos escribir la salida por cabeza como:

$$\mathbf{o}_{t,i} = \sum_{j=1}^{T} \text{Softmax}_j\left(\frac{\mathbf{q}_{t,i}^\top \mathbf{k}_{j,i}}{\sqrt{d_h}}\right)\mathbf{v}_{j,i},$$

y el vector final en el tiempo t, después de la concatenación y la proyección de salida, como:

$$\mathbf{u}_t = \mathbf{W}^O \begin{bmatrix} \mathbf{o}_{t,1} \\ \vdots \\ \mathbf{o}_{t,n_h} \end{bmatrix}.$$

Desde un punto de vista físico, la atención multi-cabeza puede leerse como **varios "canales" de interacción**: una cabeza podría enfocarse en relaciones de corto alcance, otra en relaciones de largo alcance, otra en algún patrón específico (p. ej., simetría, estructura local), y así sucesivamente.

4.1.5 Arquitectura Transformer

El **Transformer** fue introducido con el lema *"Attention Is All You Need."* [12]. Su bloque básico es una **capa** que combina **autoatención multi-cabeza** y una **red feed-forward por posición (FFN)** (ambas subcapas usan **conexiones residuales** y **normalización por capas**).

Para una secuencia de entrada $\mathbf{X} \in \mathbb{R}^{T\times d}$ (que ya incluye información posicional), una capa de Transformer realiza:

$$\mathbf{H} = \text{MHA}(\mathbf{X}), \qquad \mathbf{X}^{(1)} = \text{LayerNorm}(\mathbf{X} + \mathbf{H})\,.$$

La **subcapa feed-forward** hace (aplicada de manera independiente en cada posición):

$$\text{FFN}(\mathbf{x}) = \sigma(\mathbf{x}\mathbf{W}_1 + \mathbf{b}_1)\,\mathbf{W}_2 + \mathbf{b}_2,$$

típicamente con σ una no linealidad como ReLU o GELU y un ancho intermedio $d_{\text{ff}} > d$. A nivel de secuencia:

$$\mathbf{Z} = \text{FFN}(\mathbf{X}^{(1)}), \qquad \mathbf{X}^{\text{out}} = \text{LayerNorm}\Big(\mathbf{X}^{(1)} + \mathbf{Z}\Big)\,.$$

Apilar varias capas de este tipo da lugar a una arquitectura profunda en la que, en cada capa, cada posición puede interactuar con todas las demás posiciones mediante autoatención.

En la formulación original [12], se añaden **codificaciones posicionales** (senoidales o aprendidas) a los *embeddings* para que el modelo pueda distinguir distintas posiciones en la secuencia:

$$\mathbf{X}_0 = \mathbf{E} + \mathbf{P},$$

donde $\mathbf{E}$ son los *embeddings* de los tokens y $\mathbf{P}$ son las codificaciones posicionales.

4.1.6 ¿Por qué Transformers en lugar de RNNs o LSTMs?

Las redes neuronales recurrentes (RNNs) [45] y las redes de memoria a corto y largo plazo (LSTM) [46] procesan la secuencia de forma **secuencial**; es decir, cada nuevo estado depende del anterior. Esto tiene dos consecuencias importantes: 1. La información debe fluir a través de muchos pasos temporales, lo cual puede llevar a gradientes que se desvanecen o explotan, y hace difícil modelar interacciones de muy largo alcance. 2. Mala paralelización: como cada paso depende del anterior, no se pueden calcular todos los pasos temporales en paralelo. El entrenamiento y la inferencia son inherentemente secuenciales.

Los Transformers abordan ambos problemas: - Interacciones globales en un solo paso: la autoatención permite que cada posición interactúe directamente con cualquier otra posición en una *sola* capa, lo cual es ideal cuando nos interesan correlaciones *todo-con-todo* [12] (como en sistemas de muchos electrones, donde cada electrón "siente" a todos los demás). - Paralelismo completo sobre la longitud de la secuencia: dada $\mathbf{X}$, las matrices $\mathbf{Q}$, $\mathbf{K}$, $\mathbf{V}$ y las salidas de atención

para todos los pasos temporales se calculan mediante multiplicaciones de matrices. Esto es extremadamente eficiente en aceleradores modernos (GPUs/TPUs) [47].

Para la ecuación de Schrödinger de muchos electrones, la función de onda depende de la configuración conjunta de todas las partículas. Un ansatz basado en Transformers proporciona naturalmente una forma para que la representación de cada electrón **observe a todos los demás electrones** y a los núcleos, capturando patrones de correlación complejos mediante atención, mientras se mantiene altamente paralelizable.

5 Modelo Psi Former

5.1 Fermi Net

Un trabajo muy importante para nosotros es FermiNet [10]. Como se muestra en fig. 2, utiliza redes neuronales profundas para representar **orbitales** y luego las combina en una suma de determinantes de Slater. A nivel superior, el ansatz es una combinación lineal de K productos de determinantes:

$$\psi(\mathbf{x}_1, \dots, \mathbf{x}_n) = \sum_{k=1}^{K} \omega_k \, \det[\Phi^k], \tag{61}$$

donde ω_k son coeficientes entrenables y Φ^k es una matriz de orbitales de una partícula. Para un sistema sin separación explícita de espín, se puede escribir:

$$\det[\Phi^k] = \begin{vmatrix} \phi_1^k(\mathbf{x}_1) & \dots & \phi_1^k(\mathbf{x}_n) \\ \vdots & & \vdots \\ \phi_n^k(\mathbf{x}_1) & \dots & \phi_n^k(\mathbf{x}_n) \end{vmatrix} = \det[\phi_i^k(\mathbf{x}_j)].$$

Aquí ϕ_i^k es el i-ésimo orbital en el determinante k, y lo evaluamos en las coordenadas del electrón j.

Sin embargo, en FermiNet tratamos con electrones con espín, así que la estructura es ligeramente más rica, y los orbitales dependen de **todas** las coordenadas de los electrones, no solo de aquel en el que se "evalúan". Por eso escribimos los orbitales como:

$$\phi_i^{k\alpha}\big(\mathbf{r}_j^\alpha; \{\mathbf{r}_{/j}^\alpha\}; \{\mathbf{r}^{\bar{\alpha}}\}\big),$$

donde: - $\alpha \in \{\uparrow, \downarrow\}$ es el sector de espín, - $\mathbf{r}_j^\alpha$ es la posición del electrón j con espín α, - $\{\mathbf{r}_{/j}^\alpha\}$ denota las posiciones de todos los **otros** electrones con espín α, - $\{\mathbf{r}^{\bar{\alpha}}\}$ denota las posiciones de los electrones con el espín opuesto.

Así, el orbital evaluado en el electrón j "sabe" acerca de todos los demás electrones. El índice i corresponde al índice del orbital (fila del determinante), j al índice del electrón (columna del determinante), α, β a las etiquetas de espín ($\uparrow$ o $\downarrow$) y k al índice del determinante en la suma.

5.1.1 Coordenadas de entrada y características

Denotamos por $\mathbf{r}_1^\uparrow, \dots, \mathbf{r}_{n^\uparrow}^\uparrow$ las coordenadas de los electrones con espín arriba, $\mathbf{r}_1^\downarrow, \dots, \mathbf{r}_{n^\downarrow}^\downarrow$ las coordenadas de los electrones con espín abajo, $\mathbf{R}_I$ las posiciones de los núcleos, $I = 1, \dots, N_{\text{nuc}}$. La red construye dos tipos de características:

1. **Características electrón–núcleo** para cada electrón i con espín α:

   $$\mathbf{h}_i^{0,\alpha} = \text{concatenate}\Big(\mathbf{r}_i^\alpha - \mathbf{R}_I, \ \big|\mathbf{r}_i^\alpha - \mathbf{R}_I\big| \ \forall I\Big).$$

   Esto produce un vector de características que contiene, para el electrón (i, α), todos sus vectores de posición relativa respecto a cada núcleo, además de sus distancias.

2. **Características electrón–electrón** para cada par de electrones (i, α) y (j, β):

   $$\mathbf{h}_{ij}^{0,\alpha\beta} = \text{concatenate}\Big(\mathbf{r}_i^\alpha - \mathbf{r}_j^\beta, \ \big|\mathbf{r}_i^\alpha - \mathbf{r}_j^\beta\big| \ \forall j, \beta\Big).$$

   Para un (i, α) fijo, construimos estas características para todos los demás electrones (j, β), capturando sus posiciones relativas y distancias.

El superíndice 0 indica que estas son las características en la capa $\ell = 0$ (entrada a la red profunda). En capas más profundas seguiremos actualizando $\mathbf{h}_i^{\ell\alpha}$ (características de un solo electrón), $\mathbf{h}_{ij}^{\ell\alpha\beta}$ (características por pares), para $\ell = 0, 1, \dots, L-1$.

5.1.2 Mezcla y actualización de características a través de las capas

En cada capa oculta ℓ, queremos que las características de cada electrón dependan de *todos* los demás electrones, de una manera simétrica bajo permutaciones. Para hacer esto, formamos **promedios** sobre electrones del mismo o del espín opuesto.

Primero, definimos características globales de un solo electrón promediadas por espín:

$$\mathbf{g}^{\ell\uparrow} = \frac{1}{n^\uparrow} \sum_{j=1}^{n^\uparrow} \mathbf{h}_j^{\ell\uparrow}, \qquad \mathbf{g}^{\ell\downarrow} = \frac{1}{n^\downarrow} \sum_{j=1}^{n^\downarrow} \mathbf{h}_j^{\ell\downarrow}.$$

Luego, para cada electrón (i, α), definimos características por pares promediadas:

$$\mathbf{g}_i^{\ell\alpha\uparrow} = \frac{1}{n^\uparrow} \sum_{j=1}^{n^\uparrow} \mathbf{h}_{ij}^{\ell\alpha\uparrow}, \qquad \mathbf{g}_i^{\ell\alpha\downarrow} = \frac{1}{n^\downarrow} \sum_{j=1}^{n^\downarrow} \mathbf{h}_{ij}^{\ell\alpha\downarrow}.$$

Ahora *concatenamos* toda esta información en un único vector de características para el electrón (i, α):

$$\begin{aligned} &\big(\mathbf{h}_i^{\ell\alpha}, \frac{1}{n^\uparrow}\sum_{j=1}^{n^\uparrow} \mathbf{h}_j^{\ell\uparrow}, \frac{1}{n^\downarrow}\sum_{j=1}^{n^\downarrow} \mathbf{h}_j^{\ell\downarrow}, \frac{1}{n^\uparrow}\sum_{j=1}^{n^\uparrow} \mathbf{h}_{ij}^{\ell\alpha\uparrow}, \frac{1}{n^\downarrow}\sum_{j=1}^{n^\downarrow} \mathbf{h}_{ij}^{\ell\alpha\downarrow}\big) \\ =&\big(\mathbf{h}_i^{\ell\alpha}, \mathbf{g}^{\ell\uparrow}, \mathbf{g}^{\ell\downarrow}, \mathbf{g}_i^{\ell\alpha\uparrow}, \mathbf{g}_i^{\ell\alpha\downarrow}\big) = \mathbf{f}_i^{\ell\alpha} \end{aligned}$$

Este $\mathbf{f}_i^{\ell\alpha}$ es lo que entra a la **MLP de un solo electrón** en la capa ℓ. La actualización es:

$$\mathbf{h}_i^{\ell+1,\alpha} = \tanh\big(\mathbf{V}^\ell \mathbf{f}_i^{\ell\alpha} + \mathbf{b}^\ell\big) + \mathbf{h}_i^{\ell\alpha}$$

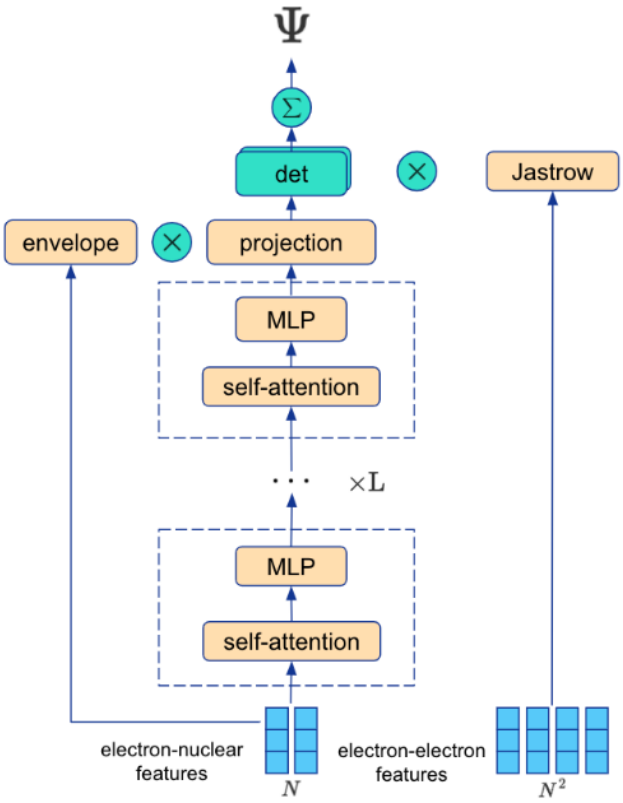

Figure 3: Psiformer utiliza un único flujo de capas de autoatención, actuando únicamente sobre características núcleo–electrón. Las características electrón–electrón aparecen solo a través del factor de Jastrow. Imagen tomada de [13].

donde $\mathbf{V}^\ell$ y $\mathbf{b}^\ell$ son pesos y sesgos entrenables, compartidos entre electrones (para el sector de espín dado). La conexión residual $+\mathbf{h}_i^{\ell\alpha}$ estabiliza el entrenamiento.

En paralelo, las características por pares se actualizan con una **MLP por pares**:

$$\mathbf{h}_{ij}^{\ell+1,\alpha\beta} = \tanh\left(\mathbf{W}^\ell \mathbf{h}_{ij}^{\ell\alpha\beta} + \mathbf{c}^\ell\right) + \mathbf{h}_{ij}^{\ell\alpha\beta},$$

con pesos $\mathbf{W}^\ell$ y sesgos $\mathbf{c}^\ell$, nuevamente compartidos sobre todos los pares (i, j, α, β).

Repitiendo estas actualizaciones para $\ell = 0, \ldots, L-1$, finalmente obtenemos **características finales de un solo electrón**:

$$\mathbf{h}_j^{L\alpha} \quad \text{para cada electrón } j \text{ de espín } \alpha.$$

Nótese cómo funcionan los índices: ℓ recorre las capas y desaparece al final; i o j siempre se refieren a un electrón específico dentro de un sector de espín; α, β indican a qué sector de espín pertenece ese electrón.

5.1.3 De las características finales a los orbitales

Los orbitales finales se construyen como una función de las características de la última capa $\mathbf{h}_j^{L\alpha}$, más algunos factores adicionales tipo "envelope" (envolvente) que manejan el decaimiento de largo alcance y las condiciones de cúspide. Para cada índice de determinante k, espín α, índice orbital i y electrón j definimos:

$$\phi_i^{k\alpha}\left(\mathbf{r}_j^\alpha; \{\mathbf{r}_{/j}^\alpha\}; \{\mathbf{r}^{\bar{\alpha}}\}\right) = \left(\mathbf{w}_i^{k\alpha} \cdot \mathbf{h}_j^{L\alpha} + g_i^{k\alpha}\right) \times \sum_m \pi_{im}^{k\alpha} \exp\Big(- \left|\mathbf{\Sigma}_{im}^{k\alpha}\left(\mathbf{r}_j^\alpha - \mathbf{R}_m\right)\right| \Big)$$

Aquí $\mathbf{w}_i^{k\alpha}$ y $g_i^{k\alpha}$ son parámetros lineales entrenables para la "parte MLP" del orbital; la suma sobre m es una "envolvente" sobre núcleos (o centros); $\pi_{im}^{k\alpha}$ y $\mathbf{\Sigma}_{im}^{k\alpha}$ son coeficientes y matrices entrenables que controlan el decaimiento exponencial alrededor del núcleo m.

Todos estos parámetros dependen de los índices: k selecciona qué determinante en la suma, i selecciona qué orbital (fila en el determinante), α selecciona el sector de espín, y m selecciona qué centro nuclear en la envolvente.

La dependencia respecto a todos los demás electrones está oculta dentro de $\mathbf{h}_j^{L\alpha}$, que fue construida a partir del conjunto completo de posiciones $\{\mathbf{r}^\uparrow\}, \{\mathbf{r}^\downarrow\}$ mediante la red profunda.

5.1.4 Ensamblando los determinantes separados por espín

Para cada índice de determinante k y sector de espín $\alpha \in \{\uparrow, \downarrow\}$, construimos una matriz:

$$D_{ij}^{k\alpha} = \phi_i^{k\alpha}\left(\mathbf{r}_j^\alpha; \{\mathbf{r}_{/j}^\alpha\}; \{\mathbf{r}^{\bar{\alpha}}\}\right),$$

con filas indexadas por la etiqueta orbital $i = 1, \ldots, n^\alpha$, y columnas indexadas por la etiqueta del electrón $j = 1, \ldots, n^\alpha$ (con ese espín).

Tomar el determinante produce una función propiamente antisimétrica de las posiciones de los electrones **con ese espín**:

$$\det\left[D^{k\alpha}\right] = \det\left[\phi_i^{k\alpha}(\mathbf{r}_j^\alpha; \{\mathbf{r}_{/j}^\alpha\}; \{\mathbf{r}^{\bar{\alpha}}\})\right].$$

Para la función de onda completa, combinamos los bloques de espín arriba y espín abajo:

$$\psi(\mathbf{r}_1^\uparrow, \ldots, \mathbf{r}_{n^\uparrow}^\uparrow, \mathbf{r}_1^\downarrow, \ldots, \mathbf{r}_{n^\downarrow}^\downarrow) = \sum_k \omega_k [D^{k\downarrow}][D^{k\uparrow}] \quad (62)$$

Aún no hemos explicado por qué podemos escribir el determinante como ese producto. En estructura electrónica, cuando se separan las partes de espín y espacial usando espín-orbitales, el determinante de Slater total sobre todos los electrones se factoriza como el producto de: un determinante que involucra solo electrones con espín arriba, y otro determinante que involucra solo electrones con espín abajo.

Cada uno de estos determinantes es antisimétrico bajo el intercambio de dos electrones **con el mismo espín**. La función de onda total construida como el producto de un determinante de espín arriba y uno de espín abajo es antisimétrica bajo el intercambio de cualesquiera dos electrones (cuando se toman en cuenta las etiquetas de espín). FermiNet mantiene esta estructura y permite que cada bloque sea representado por un ansatz de red neuronal potente para los orbitales.

Hasta este punto, los bloques de construcción son solo capas MLP (con conexiones residuales y una mezcla especial de características), pero el indexado cuidadoso (i, α) para "qué electrón/espín", j para la sumatoria sobre electrones, ℓ para las capas, y k para el índice del determinante, es lo que garantiza que el objeto final tenga la simetría por permutaciones y la antisimetría correctas requeridas para una función de onda fermiónica.

5.2 Aplicando atención a FermiNet (estilo Psiformer)

5.2.1 Factor de Jastrow para PsiFormer

La función de onda de Psiformer tiene el ansatz usual de Slater–Jastrow [30]:

$$\Psi_\theta(\mathbf{x}) = \exp\left(\mathcal{J}_\theta(\mathbf{x})\right) \sum_{k=1}^{N_{\text{det}}} \det[\mathbf{\Phi}_\theta^k(\mathbf{x})] \quad (63)$$

donde $\mathbf{x} = (x_1, \ldots, x_N)$ es la colección de los N estados electrónicos $x_i = (\mathbf{r}_i, \sigma_i)$, con $\mathbf{r}_i \in \mathbb{R}^3$ y $\sigma_i \in \{\uparrow, \downarrow\}$. Aquí, $\mathcal{J}_\theta : (\mathbb{R}^3 \times \{\uparrow, \downarrow\})^N \to \mathbb{R}$ es el **factor de Jastrow**, que codifica (en este caso) únicamente información de cúspide electrón–electrón, y $\mathbf{\Phi}_\theta^k$ es la matriz de (espín-)orbitales para el determinante k.

En Psiformer, el factor de Jastrow es *muy* simple: tiene solo dos parámetros entrenables, uno para pares con espines paralelos y otro para pares con espines antiparalelos:

$$\mathcal{J}_\theta(\mathbf{x}) = \sum_{i<j;\, \sigma_i = \sigma_j} -\frac{1}{4} \frac{\alpha_{\text{par}}^2}{\alpha_{\text{par}} + |\mathbf{r}_i - \mathbf{r}_j|} + \sum_{i,j;\, \sigma_i \neq \sigma_j} -\frac{1}{2} \frac{\alpha_{\text{anti}}^2}{\alpha_{\text{anti}} + |\mathbf{r}_i - \mathbf{r}_j|}. \quad (64)$$

donde α_{par} controla la intensidad del Jastrow para pares de electrones de **igual espín** y α_{anti} hace lo mismo para pares de **espín opuesto**.

Este Jastrow es el responsable de imponer las condiciones de cúspide electrón–electrón. La red neuronal en sí (el Psiformer) solo ve información **electrón–núcleo** en su flujo de atención; toda la dependencia explícita en $|\mathbf{r}_i - \mathbf{r}_j|$ vive en $\mathcal{J}_\theta$.

Conceptualmente, como se muestra en fig. 3, Psiformer es "FermiNet con el flujo de dos electrones reemplazado por autoatención". Solo usa características **electrón–núcleo** (más el espín) como entrada a la pila de atención. Para cada electrón i: 1. Sea $\mathbf{R}_I$ el conjunto de posiciones nucleares. 2. Construir características crudas concatenando, para todo I: alguna función de $\mathbf{r}_i - \mathbf{R}_I$ (posición relativa), $|\mathbf{r}_i - \mathbf{R}_I|$ (distancia), y el espín como un escalar (p. ej., $\sigma_i = +1$ para $\uparrow$, -1 para $\downarrow$).

En el artículo, reescalan los vectores electrón–núcleo para que las grandes distancias crezcan solo logarítmicamente [13], pero al nivel de notación podemos simplemente escribir:

$$\mathbf{f}_i^0 \in \mathbb{R}^{d_{\text{in}}} \quad \text{(características electrón–núcleo + espín)}.$$

Estas luego se mapean a la dimensión oculta del modelo mediante una capa lineal:

$$\mathbf{h}_i^0 = \mathbf{W}^0 \mathbf{f}_i^0,$$

donde $\mathbf{W}^0 \in \mathbb{R}^{d \times d_{\text{in}}}$ se aprende. Así: el índice i indica "qué electrón", y el superíndice 0 significa "antes de cualquier capa de atención".

En la capa ℓ, tenemos todos los estados ocultos electrónicos:

$$\mathbf{h}_1^\ell, \ldots, \mathbf{h}_N^\ell.$$

Para cada **cabeza** h y electrón i, calculamos:

- Query:
$$\mathbf{q}_i^{\ell h} = \mathbf{W}_q^{\ell h} \mathbf{h}_i^\ell$$
- Key:
$$\mathbf{k}_i^{\ell h} = \mathbf{W}_k^{\ell h} \mathbf{h}_i^\ell$$
- Value:
$$\mathbf{v}_i^{\ell h} = \mathbf{W}_v^{\ell h} \mathbf{h}_i^\ell$$

Aquí, cada una de las matrices $\mathbf{W}_q^{\ell h}, \mathbf{W}_k^{\ell h}, \mathbf{W}_v^{\ell h}$ es una matriz aprendida, compartida entre todos los electrones i, pero específica de la capa ℓ y de la cabeza h.

Entonces, la **salida de autoatención para el electrón i, cabeza h** es:

$$\mathbf{A}_i^{\ell h} = \sum_{j=1}^{N} \underbrace{\frac{\exp\left((\mathbf{q}_i^{\ell h})^\top \mathbf{k}_j^{\ell h}/\sqrt{d_k}\right)}{\sum_{j'=1}^{N} \exp\left((\mathbf{q}_i^{\ell h})^\top \mathbf{k}_{j'}^{\ell h}/\sqrt{d_k}\right)}}_{\text{peso de atención de } i \text{ hacia } j} \mathbf{v}_j^{\ell h}.$$

- j recorre "todos los demás electrones", así que el electrón i "mira" a todos los otros mediante atención.
- d_k es la dimensión de clave/consulta (usualmente $d_k = d/H$ o algo similar).

Esto es exactamente:

$$A_h^\ell = [\text{SelfAttn}(\mathbf{h}_1^\ell, \ldots, \mathbf{h}_N^\ell; \mathbf{W}_q^{\ell h}, \mathbf{W}_k^{\ell h}, \mathbf{W}_v^{\ell h})],$$

pero ahora escrito explícitamente con los índices i y j.

Luego, **concatenamos sobre cabezas** para cada electrón:

$$\mathbf{A}_i^\ell = \text{concat}_{h=1}^{H}\left[\mathbf{A}_i^{\ell h}\right] \in \mathbb{R}^{H d_v},$$

donde d_v es la dimensión de valores de cada cabeza.

5.2.2 Proyección residual y MLP

A continuación, mapeamos la salida de atención concatenada de vuelta a la dimensión oculta y añadimos una conexión residual:

$$\mathbf{f}_i^{\ell+1} = \mathbf{h}_i^\ell + \mathbf{W}_o^\ell \, \mathbf{A}_i^\ell,$$

donde $\mathbf{W}_o^\ell$ es una matriz aprendida.

Luego pasamos esto por una MLP pequeña, nuevamente con un residual:

$$\mathbf{h}_i^{\ell+1} = \mathbf{f}_i^{\ell+1} + \tanh\left(\mathbf{W}^{\ell+1}\mathbf{f}_i^{\ell+1} + \mathbf{b}^{\ell+1}\right).$$

Así, una capa completa de Psiformer ℓ es:

1. Autoatención: $\{\mathbf{h}_i^\ell\} \to \{\mathbf{A}_i^\ell\}$.
2. Lineal + residual: $\{\mathbf{A}_i^\ell\} \to \{\mathbf{f}_i^{\ell+1}\}$.
3. MLP + residual: $\{\mathbf{f}_i^{\ell+1}\} \to \{\mathbf{h}_i^{\ell+1}\}$.

Repetimos esto para $\ell = 0, \ldots, L-1$ y obtenemos **estados ocultos finales**:

$$\mathbf{h}_j^L \quad \text{para cada electrón } j.$$

5.2.3 De los estados ocultos a orbitales y determinantes

A partir de los estados ocultos finales $\mathbf{h}_j^L$, construimos la matriz de espín-orbitales para cada determinante k.

Para cada índice de determinante k e índice orbital i, definimos una **"cabeza orbital" lineal**:

$$\tilde{\phi}_i^k(x_j) = \mathbf{w}_i^k \cdot \mathbf{h}_j^L + g_i^k,$$

donde $\mathbf{w}_i^k$ y g_i^k son aprendidos. La dependencia del espín σ_j y de todos los demás electrones es implícita en $\mathbf{h}_j^L$: las capas de autoatención ya han mezclado esa información.

Luego multiplicamos por una **envolvente** para imponer el decaimiento asintótico correcto:

$$\Omega_{ij}^k = \sum_m \pi_{im}^k \exp\left(-\left|\mathbf{\Sigma}_{im}^k(\mathbf{r}_j - \mathbf{R}_m)\right|\right),$$

donde m indexa núcleos (o "centros de envolvente"), π_{im}^k y $\mathbf{\Sigma}_{im}^k$ son parámetros aprendidos. Las entradas finales de espín-orbital son:

$$\Phi_{ij}^k = \Omega_{ij}^k \, \tilde{\phi}_i^k(x_j).$$

Reuniendo esto en la matriz:

$$\mathbf{\Phi}^k(\mathbf{x}) = \left[\Phi_{ij}^k\right]_{i,j=1}^{N},$$

formamos el determinante:

$$\det[\mathbf{\Phi}^k(\mathbf{x})] = \det\left[\Phi_{ij}^k\right] = \det\left[\phi_i^k(x_j)\right],$$

y finalmente la función de onda completa de Psiformer:

$$\Psi_\theta(\mathbf{x}) = \exp(\mathcal{J}_\theta(\mathbf{x})) \sum_{k=1}^{N_{\text{det}}} \det[\mathbf{\Phi}_\theta^k(\mathbf{x})].$$

Las capas de autoatención son las que permiten que $\mathbf{h}_j^L$ dependa de todos los demás electrones de una forma flexible y aprendida, mientras que el determinante sobre i, j y el Jastrow sobre i, j imponen la antisimetría fermiónica y las condiciones de cúspide.

6 Metodología

La implementación de PsiFormer requiere un marco computacional capaz de expresar arquitecturas neuronales flexibles y, al mismo tiempo, ser eficiente en hardware paralelo a gran escala. Los ecosistemas modernos de aprendizaje profundo ofrecen varias posibilidades, en particular JAX, TensorFlow y PyTorch. Cada uno proporciona diferenciación automática y soporte para GPU/TPU. PyTorch, ampliamente adoptado en dominios de investigación, ofrece un grafo de cómputo dinámico y un ecosistema rico de herramientas científicas. Trabajos previos en Variational Monte Carlo, incluyendo FermiNet, han motivado principios de diseño similares, aunque su implementación original se basó en TensorFlow [14, 13, 10]. Dado el mayor apoyo de la comunidad, la simplicidad para depurar y la disponibilidad de implementaciones de código abierto de arquitecturas basadas en atención, se elige PyTorch como el marco principal.

El entrenamiento de modelos de la familia PsiFormer requiere recursos computacionales significativos, como reportan los autores originales. Las funciones de onda de estructura electrónica demandan muestreo repetido, evaluaciones de determinantes costosas y horizontes de optimización largos, llevando los tiempos de entrenamiento típicos a varios días o semanas dependiendo de la complejidad molecular.

Para satisfacer estos requisitos, utilizamos entornos acelerados por GPU con soporte CUDA. Aunque plataformas de notebooks en la nube como Google Colab ofrecen GPUs accesibles, su inestabilidad, límites de sesión y tiempos de ejecución restringidos las hacen inadecuadas para experimentos de larga duración. En su lugar, el proyecto utiliza instancias de GPU alquiladas (por ejemplo, RunPod o proveedores comparables).

7 Resultados

En esta sección presentamos los resultados numéricos obtenidos con el ansatz propuesto (PsiFormer) entrenado mediante VMC. El objetivo principal es evaluar: (i) precisión energética en estados fundamentales, (ii) estabilidad/eficiencia del entrenamiento, y (iii) el efecto de las decisiones arquitectónicas (autoatención, Jastrow, y/o preacondicionamiento tipo gradiente natural/KFAC).

7.1 Configuración experimental

Entrenamos el modelo minimizando el cociente de Rayleigh (energía esperada) usando el estimador de Monte Carlo de la energía local y muestreo MCMC con Metropolis–Hastings, tal como se describió en eq. (38). Para cada sistema reportamos:

- Número de caminantes (walkers) N_w, pasos de *burn-in* N_{eq} y tamaño del paso (o desviación estándar de la propuesta gaussiana) σ_{MH}.
- Número de muestras efectivas por iteración M y tasa de aceptación promedio.
- Arquitectura: dimensión oculta d, número de capas L, número de cabezas H, y número de determinantes N_{det}.
- Optimizador: (i) primer orden (p. ej. Adam) o (ii) preacondicionado (p. ej. gradiente natural/KFAC).

Nota: para reproducibilidad, recomendamos fijar semillas, registrar la tasa de aceptación MH, y reportar intervalos de confianza (por ejemplo, error estándar) calculados con *blocking* o estimación de autocorrelación.

7.2 Energías de estado fundamental

La métrica primaria es la energía del estado fundamental E_0 (en Hartree) estimada por VMC:

$$\widehat{E}_0 \;=\; \frac{1}{M}\sum_{k=1}^{M} E_L(\mathbf{R}_k), \qquad \mathbf{R}_k \sim p_\theta(\mathbf{R}).$$

En table 1 comparamos PsiFormer con baselines típicos (HF/DFT, y/o modelos tipo FermiNet cuando estén disponibles) y con valores de referencia cuando corresponda.

Table 1: Energías de estado fundamental.

Sistema	N_e	Método	Energía (Ha)	Error vs ref
H	1	Psiformer T.	- 0.499	0.001
He	2	PsiFormer T.	- 4.801 0.	0.103
Li	3	PsiFormer T.	-7.423	0.353
Be	4	PsiFormer T.	-14.167	0.557
B	5	PsiFormer T.	-23.612	1.393
C	6	PsiFormer T.	-36.367	1.523

7.3 Curvas de convergencia y estabilidad

Además de la energía final, analizamos el comportamiento dinámico del entrenamiento: (i) convergencia de $\mathbb{E}[E_L]$, (ii) varianza de la energía local $\mathrm{Var}(E_L)$ (indicador de calidad del ansatz y del muestreo), y (iii) estabilidad numérica al trabajar en log-espacio.

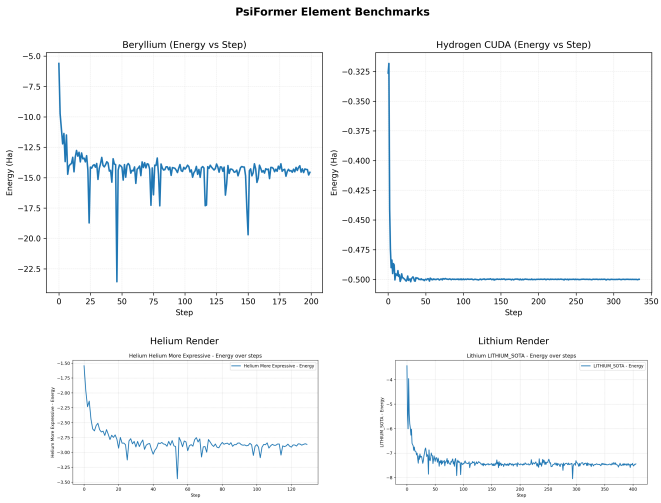

Figure 4: Curva típica de entrenamiento: energía estimada (y barras de incertidumbre) vs iteraciones. Sustituir por resultados.

8 Análisis y discusión

8.1 Interpretación física: correlaciones y simetrías

El punto central de PsiFormer es que la autoatención implementa una mezcla *todo-con-todo* entre electrones, en paralelo, y de forma permutacionalmente equivariante/invariante según la construcción. Desde el punto de vista físico, esto equivale a permitir que cada electrón construya una representación contextual que depende del resto de la configuración electrónica, capturando correlaciones no triviales (incluyendo correlación dinámica y parte de la correlación estática) sin introducir explícitamente términos por pares en el flujo principal, ya que la dependencia $|\mathbf{r}_i - \mathbf{r}_j|$ se concentra en el factor de Jastrow.

En particular:

- **Antisimetría:** está garantizada por la(s) determinante(s) de Slater; la atención modifica los orbitales efectivos, no la estructura antisimétrica fundamental.
- **Cúspides:** la inclusión explícita de $\mathcal{J}_\theta$ alivia la carga de la red, al imponer condiciones de cúspide electrón–electrón de manera analítica (o al menos controlada).
- **Largo alcance:** los factores *envelope* fuerzan el decaimiento correcto, reduciendo extrapolaciones no físicas a grandes distancias.

8.2 Limitaciones

Aun cuando PsiFormer ofrece una parametrización flexible, hay límites claros:

- **Escalamiento:** la autoatención estándar es $O(N^2)$ en número de electrones, lo cual puede volverse costoso para sistemas grandes.
- **Dependencia del muestreo:** un muestreo pobre puede dominar el error total, independientemente de la capacidad del modelo.
- **Referencias:** para validar precisión química se requieren referencias confiables (p. ej. DMC/FCI cuando sea viable) y reportes sistemáticos.

9 Conclusiones

En este trabajo presentamos y analizamos un ansatz tipo PsiFormer para la ecuación de Schrödinger de muchos electrones, combinando:

- una pila de **autoatención** que mezcla información electrónica de forma global y paralela,
- un **determinante de Slater** (o suma de determinantes) que garantiza la **antisimetría fermiónica**,
- un **factor de Jastrow** simple que impone (o refuerza) **condiciones de cúspide electrón–electrón**,
- y entrenamiento mediante **Monte Carlo variacional** (con opción de **gradiente natural/KFAC** para mejorar la eficiencia).

Los resultados (ver table 1 and fig. 4) muestran que el enfoque es competitivo para estados fundamentales en sistemas pequeños a medianos, y que la combinación "atención + Jastrow + determinantes" permite capturar correlaciones electrónicas relevantes con buena estabilidad numérica.

Como trabajo futuro, las direcciones más prometedoras incluyen: (i) estudiar variantes de atención más eficientes (sparse/linear) para mejorar escalamiento, (ii) enriquecer el Jastrow o imponer cúspides electrón–núcleo de forma más explícita, (iii) evaluar sistemáticamente moléculas más grandes y superficies de energía potencial, y (iv) comparar en igualdad de cómputo con baselines consolidados (FermiNet y variantes recientes).

En resumen: la autoatención ofrece un lenguaje natural para las correlaciones *todo-con-todo* propias de sistemas de muchos electrones, y al integrarse con las restricciones físicas correctas (antisimetría y cúspides) produce un ansatz potente, entrenable y alineado con la estructura del problema cuántico.

References

[1] Erwin Schrödinger. An undulatory theory of the mechanics of atoms and molecules. 28(6):1049–1070, 1926.

[2] W. Kohn and L. J. Sham. Self-consistent equations including exchange and correlation effects. *Physical Review*, 140(4A):A1133–A1138, 1965.

[3] M. Born and R. Oppenheimer. Zur quantentheorie der molekeln. 84(20):457–484, 1927.

[4] W. L. McMillan. Ground state of liquid he^4. *Physical Review*, 138(A442–A447):A442–A447, 1965.

[5] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Anna Žıdek, Anna Potapenko, et al. Highly accurate protein structure prediction with AlphaFold. 596(7873):583–589, 2021.

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.

[7] M. Raissi, P. Perdikaris, and G.E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. 378:686–707, 2019.

[8] Di Luo and Bryan K. Clark. Backflow transformations via neural networks for quantum many-body wave functions. *Physical Review Letters*, 122(22), 2019.

[9] Zhuoran Qiao, Matthew Welborn, Animashree Anandkumar, Frederick R. Manby, and Thomas F. Miller. OrbNet: Deep learning for quantum chemistry using symmetry-adapted atomic-orbital features. 153(12), 2020-09.

[10] David Pfau, James S. Spencer, Alexander G. D. G. Matthews, and W. M. C. Foulkes. Ab initio

solution of the many-electron Schrödinger equation with deep neural networks. 2(3), 2020-09.

[11] Jan Hermann, Zeno Schätzle, and Frank Noé. Deep-neural-network solution of the electronic schrödinger equation. *Nature Chemistry*, 12(10):891–897, 2020.

[12] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. 30, 2017.

[13] James S. Glehn, Ingrid von Spencer and David Pfau. A self-attention ansatz for ab-initio quantum chemistry, 2023.

[14] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. arXiv preprint, 2019.

[15] Niels Bohr. The quantum postulate and the recent development of atomic theory. *Nature*, 121(3050):580–590, 1928.

[16] Louis de Broglie. *Recherches Sur La Théorie Des Quanta*. PhD thesis, Université de Paris, 1924.

[17] Nouredine Zettili. *Quantum Mechanics: Concepts and Applications*. John Wiley & Sons, Chichester, UK, 2 edition, 2009.

[18] S. N. Bose. Planck's law and the light quantum hypothesis. *Zeitschrift fr Physik*, 26:178–181, 1924.

[19] Albert Einstein. Quantentheorie des einatomigen idealen gases. *Sitzungsberichte der Preuischen Akademie der Wissenschaften, Physikalisch-mathematische Klasse*, pages 261–267, 1924.

[20] Enrico Fermi. Zur quantelung des idealen einatomigen gases. *Zeitschrift für Physik*, 36:902–912, 1926.

[21] Paul Adrien Dirac. On the theory of quantum mechanics. *Proceedings of the Royal Society of London. Series A*, 112:661–677, 1926.

[22] Wolfgang Pauli. Über den zusammenhang des abschlusses der elektronengruppen im atom mit der komplexstruktur der spektren. *Zeitschrift für Physik*, 31(1):765–783, 1925.

[23] John C. Slater. The theory of complex spectra. *Physical Review*, 34:1293–1322, 1929.

[24] T. Kato. On the eigenfunctions of many-particle systems in quantum mechanics. *Communications on Pure and Applied Mathematics*, 10(2):151–177, 1957.

[25] Robert Jastrow. Many-body problem with strong forces. *Physical Review*, 98(5):1479–1484, 1955.

[26] Particle Data Group. Review of particle physics. *Progress of Theoretical and Experimental Physics*, 2(8):083C01, 2024.

[27] Isaac Newton. *Philosophiae Naturalis Principia Mathematica*. Joseph Streater, London, 1687.

[28] Walther Ritz. Über eine neue methode zur lösung gewisser variationsprobleme der mathematischen physik. *Journal für die reine und angewandte Mathematik*, 135:1–61, 1909.

[29] Herbert Goldstein, Charles P. Poole, and John L. Safko. *Classical Mechanics*. Addison-Wesley, 3 edition, 2002.

[30] W. M. C. Foulkes, L. Mitas, R. J. Needs, and G. Rajagopal. Quantum monte carlo simulations of solids. *Rev. Modern Phys.*, 73:33, 2001.

[31] M. Bajdich, L. Mitas, G. Drobný, L. K. Wagner, and K. E. Schmidt. Pfaffian pairing wave functions in electronic-structure quantum monte carlo simulations. *Phys. Rev. Lett.*, 96:130201, 2006.

[32] Nicholas Metropolis and Stanislaw Ulam. The monte carlo method. *Journal of the American Statistical Association*, 44(247):335–341, 1949.

[33] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21(6):1087–1092, 1953.

[34] David Donoho. High-dimensional data analysis: The curses and blessings of dimensionality. *AMS Math Challenges Lecture*, pages 1–32, 2000.

[35] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

[36] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[37] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

[38] Michael A. Nielsen. *Neural Networks and Deep Learning*. Self-published, 2015.

[39] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006.

[40] Solomon Kullback and Richard A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22(1):79–86, 1951.

[41] Ronald A. Fisher. On the mathematical foundations of theoretical statistics. *Philosophical Transactions of the Royal Society of London. Series A*, 222:309–368, 1922.

[42] Shun-ichi Amari. Natural gradient works efficiently in learning. 10:251–276, 1998.

[43] James Martens and Roger Grosse. Optimizing neural networks with kronecker-factored approximate curvature. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, volume 37 of *Proceedings of Machine Learning Research*, pages 2408–2417, Lille, France, July 2015. PMLR.

[44] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. 2014.

[45] Jeffrey L Elman. Finding structure in time. 14(2):179–211, 1990.

[46] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. 9(8):1735–1780, 1997.

[47] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models, 2020.