

In-Context Imitation Learning for UAV Navigation via Graph Diffusion Model

Miguel Augusto Azar¹

¹ Universidad de Buenos Aires, Buenos Aires, Argentina
mazar@fi.uba.ar

Abstract. This work presents an adaptation of Instant Policy (Vosylius & Johns, 2025), an in-context imitation learning (ICIL) framework based on graph diffusion, to the domain of unmanned aerial vehicle (UAV) navigation using the Crazyflie 2.1 quadrotor as reference platform (Bitcraze AB, 2023). The original work addresses robotic manipulation tasks with a gripper; the contribution of this work extends the approach to four navigation tasks: fly to a marker, inspect an object, land on a platform, and obstacle avoidance with goal reaching. The heterogeneous graph representation is reformulated to accommodate the drone state (position and orientation) and objects in the 3D environment. A pseudo-demonstration generator with simplified dynamics, whose parameters are partially based on the public specifications of the Crazyflie 2.1, is introduced as a training data source. Results show that Instant Policy achieves minimum distances to the goal equivalent to those of the Behavioral Cloning (BC) baseline across all tasks, while producing qualitatively richer trajectories. Limitations and future directions are discussed.

Keywords: imitation learning, graph neural networks, diffusion models, drone navigation, in-context learning.

Aprendizaje por Imitación en Contexto para Navegación de VANT mediante Modelo de Difusión en Grafos

Resumen. Este trabajo presenta una adaptación de Instant Policy (Vosylius & Johns, 2025), un marco de aprendizaje por imitación en contexto (ICIL) basado en difusión sobre grafos, al dominio de navegación de vehículos aéreos no tripulados (VANT) utilizando como referencia el cuadricóptero Crazyflie 2.1 (Bitcraze AB, 2023). El trabajo original aborda tareas de manipulación robótica con gripper; la contribución de este trabajo extiende el enfoque a cuatro tareas de navegación: volar a un marcador, inspeccionar objeto, aterrizar en plataforma y evasión de obstáculos con alcance de objetivo. Se reformula la representación de grafo heterogéneo para acomodar el estado del dron (posición y orientación) y los objetos del entorno 3D. Se introduce un generador de pseudodemostraciones con dinámica simplificada, cuyos parámetros se basan parcialmente en las especificaciones públicas del Crazyflie 2.1, como fuente de datos de entrenamiento. Los resultados muestran que Instant Policy alcanza distancias mínimas al objetivo equivalentes a las del baseline de Behavioral Cloning (BC) en todas las tareas, produciendo trayectorias cualitativamente más ricas. Se discuten limitaciones y direcciones futuras.

Palabras clave: aprendizaje por imitación, redes neuronales en grafos, modelos de difusión, navegación de drones, aprendizaje en contexto.

1 Introducción

Enseñar nuevas tareas a los robots de forma eficiente sigue siendo un desafío fundamental en robótica y aprendizaje automático. Los enfoques tradicionales, como la clonación del comportamiento (BC), codifican el conocimiento de la tarea directamente en los pesos de la red, lo que requiere un reentrenamiento para cada nueva tarea o configuración del entorno. El aprendizaje por imitación en contexto ofrece una alternativa: la política se adapta a nuevas tareas en tiempo de inferencia utilizando un pequeño número de demostraciones como contexto, sin modificar sus parámetros.

Instant Policy (Vosylius & Johns, 2025) es un marco ICIL reciente que logra esto mediante dos componentes: una representación heterogénea en forma de grafo que codifica conjuntamente demostraciones, observaciones actuales y acciones futuras; y un proceso de difusión sobre ese grafo que genera acciones mediante la eliminación iterativa de ruido gaussiano condicionado a la estructura del grafo. Si bien el trabajo original centra la implementación de este método en la manipulación robótica, el paradigma ICIL podría ser aplicable en otros ámbitos. La navegación de VANT presenta una extensión plausible: los drones deben adaptarse a entornos variables, posiciones de objetos y especificaciones de tareas, a menudo con datos de demostración limitados. En este trabajo se investiga si los mecanismos centrales de Instant Policy son transferibles al dominio de los VANT. Las contribuciones son: (1) una reformulación

de la representación en grafo para la navegación de drones con 4 grados de libertad; (2) un generador de pseudodemostraciones con dinámica simplificada basada parcialmente en parámetros del Crazyflie 2.1; (3) una comparación empírica con una línea base de BC en cuatro tareas de navegación; y (4) la identificación de limitaciones relevantes y direcciones para la mejora.

2 Método

2.1 Representación en grafo

Siguiendo el marco original, se construye un grafo heterogéneo con cinco tipos de nodos: dron, objeto, acción, estado del contexto y acción del contexto.

El nodo dron codifica la pose actual como un vector de 9 dimensiones: las primeras 3 corresponden a la posición (x, y, z) normalizada respecto al tamaño del entorno para mantener los valores en un rango uniforme; las 6 restantes codifican la orientación del dron. Dado que representar la orientación directamente con un cuaternión introduce ambigüedades numéricas que dificultan el aprendizaje, se utilizan en su lugar las primeras dos columnas de la matriz de rotación 3×3 equivalente, lo que produce 6 valores sin ambigüedad.

Los nodos de objeto codifican 11 dimensiones: 3 de posición (x, y, z) normalizada, 3 de tamaño del cuadro delimitador (ancho, largo, alto) que aproxima la geometría del objeto, y 5 de un vector de tipo one-hot. Este último indica de cuál de los 5 posibles tipos de objeto se trata (marcador, objeto de inspección, plataforma, obstáculo u objetivo) usando una codificación binaria donde una sola posición vale 1 y el resto valen 0, evitando así imponer una jerarquía numérica inexistente entre los tipos.

Los nodos de acción representan las 8 acciones futuras a predecir, cada una como el vector $(\Delta_x, \Delta_y, \Delta_z, \Delta_{yaw}) \in \mathbb{R}^4$, donde cada componente es un desplazamiento incremental respecto a la pose actual. Predecir un horizonte de 8 pasos en lugar de una sola acción le permite al modelo planificar trayectorias coherentes; en ejecución, el dron aplica únicamente la primera acción del horizonte y repite el proceso de inferencia con la nueva observación.

Los tipos de aristas codifican las siguientes relaciones semánticas: relaciones de percepción entre el par (dron, objeto), relaciones de control entre (dron, acción), relaciones de influencia de la escena entre (objeto, acción) y relaciones de guía en contexto entre (estado-contexto, acción). Este último tipo de arista constituye la vía principal a través de la cual el conocimiento contenido en las demostraciones de contexto condiciona la predicción de acciones. Cabe destacar que la arquitectura presentada aquí difiere del trabajo original en que los nodos de contexto se integran directamente en el grafo de observación, en lugar de procesarse mediante un codificador de contexto separado, lo que representa una simplificación con implicaciones sobre el rendimiento discutidas en la Sección 4.

2.2 Arquitectura del modelo

El modelo consta de dos componentes. El primero es un codificador GNN heterogéneo con 6 capas SAGEConv (Hamilton et al., 2017; Fey & Lenssen, 2019), implementado mediante HeteroConv en PyTorch Geometric, con dimensión oculta $h = 256$, es decir, 256 neuronas en cada capa oculta de la red, lo que resulta en un total de 4,9 millones de parámetros entrenables. El segundo componente es un modelo de eliminación de ruido DDPM (Ho et al., 2020) que predice el ruido a remover en cada paso de difusión, condicionado tanto a las características propagadas por el grafo como a una representación vectorial del paso de tiempo actual, codificada mediante funciones sinusoidales de distintas frecuencias. El esquema de ruido β lineal determina cuánto ruido gaussiano se agrega en cada paso del proceso de difusión hacia adelante: va desde $\beta_1 = 10^{-4}$ hasta $\beta_{100} = 0,02$ en $T = 100$ pasos, de modo que la señal queda progresivamente dominada por ruido hasta ser irreconocible al final del proceso. El modelo aprende a revertir ese proceso.

2.3 Pseudodemostraciones basadas en dinámica simplificada

Para el entrenamiento se emplean pseudodemostraciones: trayectorias sintéticas generadas automáticamente, sin intervención humana, que sirven como fuente prácticamente ilimitada de datos de entrenamiento. Esto sigue el principio central de Instant Policy (Vosylius & Johns, 2025), donde las demostraciones no necesitan ser de alta fidelidad, sino semánticamente consistentes con la tarea.

Un simulador dinámico de primer orden genera estas trayectorias incorporando: inercia traslacional con masa $m = 0,027$ kg (valor nominal del Crazyflie 2.1 según especificaciones del fabricante), resistencia aerodinámica cuadrática con coeficiente empírico $k_d = 0,15$ (parámetro adimensional que escala la fuerza de arrastre en función del cuadrado de la velocidad), retardo de respuesta de los motores modelado como un filtro de paso bajo de primer orden con constante de tiempo $\tau = 0,08$ s (que implica que los motores tardan aproximadamente 80 ms en alcanzar el 63% de la velocidad comandada), ruido gaussiano en la posición percibida del drone (que simula la imprecisión característica de sistemas de odometría visual-inercial reales) y perturbaciones externas aleatorias con probabilidad 0,03 por paso de tiempo (que simulan ráfagas de viento u otras perturbaciones no modeladas). Las trayectorias se generan ejecutando un controlador proporcional hacia puntos de referencia específicos de cada tarea dentro de este simulador, produciendo 5.000 demostraciones por tarea (20.000 en total). Es importante señalar que las demostraciones de contexto empleadas durante la evaluación (las 1 o 2 demostraciones que definen la tarea a realizar) se generan mediante el mismo procedimiento, lo que constituye una simplificación metodológica respecto al escenario ideal de ICIL donde el contexto provendría de demostraciones humanas reales.

2.4 Entrenamiento

El modelo se entrena con el objetivo estándar de predicción de ruido de DDPM: dado un ejemplo de acción, un paso de tiempo de difusión aleatorio y la versión ruidosa de

esa acción, el modelo debe predecir el ruido que fue añadido. El entrenamiento procesa 24 ejemplos simultáneamente en cada actualización de parámetros (tamaño de lote), lo que balancea estabilidad del gradiente y uso de memoria de la GPU. Se emplea el optimizador AdamW con tasa de aprendizaje inicial de 3×10^{-4} y un esquema de recocido coseno, que reduce la tasa de aprendizaje suavemente a lo largo del entrenamiento siguiendo una curva de coseno hasta aproximarse a cero en la última época. El entrenamiento se ejecuta durante 100 épocas en una GPU NVIDIA T4 (aproximadamente 103 minutos). La pérdida de validación final es de 0,01023.

3 Experimentos

3.1 Tareas y protocolo de evaluación

Se evalúan cuatro tareas de navegación: (1) volar hacia un marcador: alcanzar un disco de color ubicado en el suelo; (2) inspeccionar un objeto: orbitar un objeto 3D a corta distancia; (3) aterrizar en una plataforma: posarse con precisión sobre una plataforma elevada; (4) esquivar y seguir: sortear un obstáculo para alcanzar un objetivo. Cada tarea se evalúa con 2 demostraciones de contexto sintéticas que definen el comportamiento deseado, y se ejecutan 10 episodios de prueba con posiciones de objetos generadas aleatoriamente para medir la capacidad de generalización. Los umbrales de éxito (distancia euclidiana al objetivo) son específicos para cada tarea: 0,30 m (volar), 0,90 m (inspeccionar, criterio de radio orbital), 0,65 m (aterrizar) y 1,20 m (esquivar y seguir). Se comparan los resultados con una línea base de BC: una red neuronal de tipo MLP con 0,5 millones de parámetros, entrenada con regresión supervisada sobre el mismo conjunto de datos y con acceso al mismo contexto.

3.2 Resultados

La Tabla 1 resume los resultados de evaluación. Instant Policy logra un 100% de éxito en las tareas de vuelo hacia el marcador y de evasión con seguimiento, y un 80% en las de inspección de objeto y aterrizaje en plataforma. Las distancias mínimas al objetivo son comparables entre ambos modelos en todas las tareas. Un hallazgo relevante es que el rendimiento de IP es altamente sensible al número de pasos de eliminación de ruido empleados durante la inferencia: reducirlo de $T = 100$ a $T = 20$ colapsa la tasa de éxito del 80–100% al 0% en todas las tareas, lo que confirma que la convergencia completa del proceso de difusión es necesaria para generar acciones coherentes.

Tabla 1. Resultados de la evaluación en 10 episodios por tarea. SR (success rate): tasa de éxito. Dist: distancia mínima media al objetivo. Negrita: mejor resultado por métrica.

Tarea	IP SR	BC SR	IP Dist	BC Dist
Volar a un marcador	100%	100%	0.289m	0.289m
Inspeccionar objeto	80%	100%	0.901m	0.882m
Aterrizar en plataforma	80%	100%	0.658m	0.642m
Esquivar y seguir	100%	100%	1.198m	1.184m

Las Figuras 1, 2 y 3 ilustran el comportamiento del modelo durante la evaluación. La Figura 1 muestra las curvas de convergencia al objetivo a lo largo de los pasos de ejecución para el primer episodio de cada tarea. Se observa que IP reduce la distancia al objetivo de forma sostenida en todas las tareas, con velocidades de convergencia que reflejan la complejidad geométrica de cada una: la tarea de vuelo al marcador converge más rápidamente, mientras que la de evasión y seguimiento requiere más pasos debido a la naturaleza de dos fases del comportamiento.

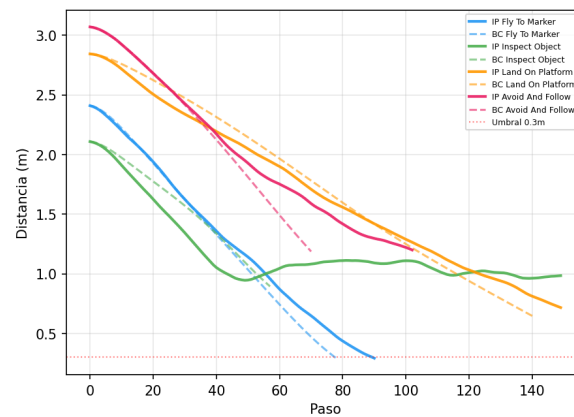


Fig. 1. Curvas de convergencia al objetivo durante la evaluación (episodio 1 por tarea). Cada curva representa la distancia euclidiana entre el drone y el objetivo a lo largo de los pasos de ejecución. La línea punteada indica el umbral de éxito de referencia de 0,30 m.

La Figura 2 presenta las trayectorias de IP en vista superior (plano XY). Es especialmente notable la trayectoria de inspección de objeto, donde emerge un patrón orbital no presente en el baseline BC: el drone rodea el objeto antes de aproximarse, comportamiento que surge directamente de las demostraciones de contexto sin ser programado explícitamente. La Figura 3 muestra el perfil de altura (eje Z) durante la ejecución. La tarea de aterrizaje exhibe claramente las dos fases del comportamiento aprendido: primero desplazamiento lateral para posicionarse sobre la plataforma, y luego descenso controlado.

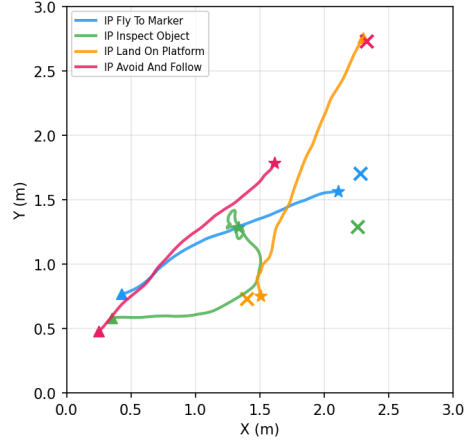


Fig. 2. Trayectorias de Instant Policy en vista superior (plano XY) para el primer episodio de cada tarea. Los triángulos indican la posición inicial, las estrellas la posición final y las cruces la ubicación del objetivo. El comportamiento orbital emergente en la tarea de inspección (verde) contrasta con las trayectorias más directas de las demás tareas.

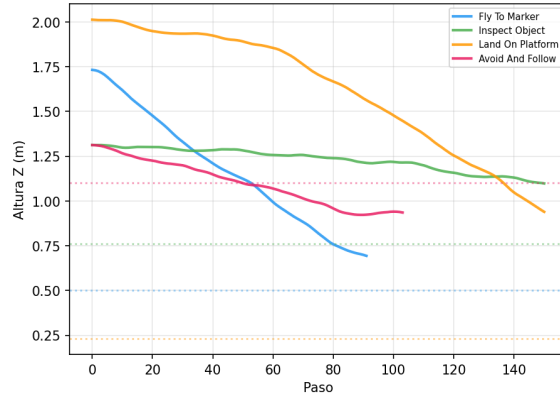


Fig. 3. Perfil de altura (eje Z) durante la ejecución de cada tarea (episodio 1). Las líneas punteadas horizontales indican la altura objetivo para cada tarea. La tarea de aterrizaje en plataforma (naranja) exhibe el comportamiento de dos fases: desplazamiento lateral seguido de descenso.

4 Discusión y Conclusiones

Los resultados demuestran que el ICIL basado en difusión sobre grafos es aplicable a la navegación de VANT. IP iguala o se aproxima al rendimiento de BC produciendo trayectorias con estructura geométrica implícita, en particular el comportamiento orbital emergente en la tarea de inspección. El baseline BC logra mayor consistencia (100% frente a 80%) en dos tareas. Esta diferencia se atribuye principalmente a la au-

sencia del codificador de contexto dedicado del trabajo original (Vosylius & Johns, 2025), que procesa cada demostración de forma independiente antes de integrarla al grafo principal. La adaptación presentada integra los nodos de contexto directamente en el grafo de observación, lo que limita la capacidad del modelo para ponderar las demostraciones según su relevancia para la etapa actual de la tarea. Una segunda limitación relevante es que las demostraciones de contexto son sintéticas, generadas por el mismo procedimiento que las pseudodemostraciones de entrenamiento, lo que difiere del escenario de uso real de ICIL.

En trabajos futuros se abordarán estas limitaciones y se escalará el desarrollo con los objetivos de: (1) implementar el codificador de contexto dedicado siguiendo la arquitectura original; (2) evaluar el eventual reemplazo del simulador de dinámica simplificada por gym-pybullet-drones (Panerati et al., 2021) para pseudodemostraciones de mayor fidelidad; (3) extender la evaluación a 100 episodios por tarea con análisis estadístico; y (4) validar en hardware físico Crazyflie mediante cámaras de profundidad externas, esto es, la validación implica contemplar el uso de un sistema de captura de movimiento para estimar las poses del dron y los objetos del entorno, transmitiendo comandos de velocidad calculados por el modelo IP a través del protocolo CRTP (Crazy RealTime Protocol) y siguiendo el esquema de control externo empleado en el trabajo original.

Este trabajo establece la viabilidad de adaptar el ICIL basado en difusión de grafos a la navegación de VANT e identifica los componentes arquitectónicos necesarios para cerrar la brecha de rendimiento con métodos más simples.

Referencias

- Fey, M. & Lenssen, J. E. (2019). Fast graph representation learning with PyTorch Geometric. ICLR Workshop on Representation Learning on Graphs and Manifolds.
- Hamilton, W., Ying, Z. & Leskovec, J. (2017). Inductive representation learning on large graphs. *Advances in Neural Information Processing Systems*, 30.
- Ho, J., Jain, A. & Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33, 6840–6851.
- Panerati, J., Zheng, H., Zhou, S., Xu, R., Negahban, S. & Schoellig, A. P. (2021). Learning to fly — a gym environment with PyBullet physics for reinforcement learning of multi-agent quadcopter control. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.
- Vosylius, V. & Johns, E. (2025). Instant policy: In-context imitation learning via graph diffusion. *International Conference on Learning Representations (ICLR)*.
- Bitcraze AB. (2023). Crazyflie 2.1 technical specifications. Bitcraze. <https://store.bitcraze.io/products/crazyflie-2-1/>, último acceso 2026-04-14.