

Utiliser, personnaliser et entraîner des IA: jusqu'où les architectes peuvent-ils/doivent-ils aller ?

Claire Duclos-Prévet

Maîtresse de conférences associée à l'ENSAPLV

Chercheuse au MAP-MAACC et pour la cellule R&D d'Architecturestudio

Associée à une possible transformation en profondeur du rôle du concepteur et à des interrogations éthiques, l'intelligence artificielle (IA) est souvent accueillie avec réserve dans le milieu de l'architecture. Les applications récentes fondées sur les grands modèles de langage (LLM) ou sur les modèles de diffusion ne couvrent pourtant qu'une fraction des possibilités offertes par l'IA.

Aujourd'hui, nous distinguons trois niveaux d'appropriation de l'IA, qui n'exigent ni les mêmes compétences ni le même investissement, et n'offrent pas le même potentiel : l'usage d'applications accessibles à tous ; la personnalisation d'outils existants, qui élargit déjà sensiblement le champ des possibles ; et enfin l'entraînement de modèles propres, qui ouvre l'accès à des terrains techniques jusqu'ici rarement investis par les architectes.

1. L'utilisation d'applications basées sur des modèles d'IA

Les modèles de diffusion sont les techniques mobilisées par la plupart des applications de génération d'images. En pratique, ils apprennent à reconstruire une image à partir d'une version progressivement bruitée, guidés par une consigne textuelle (prompt). Des outils comme Midjourney ou DALL·E rendent ces usages très accessibles. En architecture, ils peuvent soutenir la phase d'idéation, mais il faut garder à l'esprit qu'ils ne disposent pas, en tant que tels, d'une compréhension physique de l'espace, du projet, des contraintes ou du contexte. Si l'on s'en tient au seul « art du prompt », le contrôle reste limité et le text-to-image pur produit souvent des résultats difficiles à exploiter. En revanche, ces modèles deviennent particulièrement pertinents en post-production : amélioration et stylisation d'images, retouches de rendus 3D, variations d'ambiance (figure 1).

Les architectes sont amenés à lire et produire une quantité importante de documentation (programmes, réglementations, comptes rendus de réunion, notices architecturales, appels d'offres, échanges d'e-mails, etc.). Les grands modèles de langage (LLM) peuvent faire gagner un temps réel sur ces tâches, mais ils se distinguent surtout comme assistants de programmation. Pour celles et ceux qui pratiquent Grasshopper ou Dynamo (respectivement sur Rhinocéros et Revit), un LLM peut faciliter l'écriture de composants personnalisés à partir d'un cahier des charges précis — une pratique parfois qualifiée de « vibe coding ». Le gain de temps peut être considérable, à condition de conserver une capacité de vérification et de compréhension du code produit.



Figure 1 : Perspectives 3D réalisées avec les logiciels Rhinocéros et Artlantis (à gauche en 2016) + ChatGPT (à droite en 2026), diplôme de Claire Duclos-Prévet, ENSAPB

Moins connus mais plus inquiétants pour la profession, des outils de génération de plans fondés sur l'optimisation et l'apprentissage profond commencent également à apparaître, avec des performances encore inégales. Ils sont aujourd'hui particulièrement développés sur le logement (Finch 3D, Maket.ai, etc.), un domaine pour lequel existent des bases de données de grande taille (plusieurs dizaines de milliers de plans). Les données d'entraînement mobilisées par les entreprises restent toutefois souvent opaques, et les résultats peuvent s'avérer génériques ou peu robustes face à des géométries complexes, avec une adaptation limitée au contexte. Néanmoins, la recherche progressant rapidement sur ces approches, leur impact pourrait croître à court terme.

2. La personnalisation de modèles d'IA pré-entraînés

La personnalisation de modèles d'IA, accessible aux plus motivés, offre davantage de flexibilité et de contrôle sur le processus génératif. Elle consiste à alimenter un modèle pré-entraîné avec des données personnelles.

Pour les images, des modèles de diffusion comme Stable Diffusion (SD) peuvent être spécialisés pour l'architecture via des checkpoints publiés en libre accès (par exemple sur Civitai). Mais surtout, il est possible d'entraîner des LoRA, Low-Rank Adaptation, (Hu et al., 2021), qui ajoutent une couche légère de personnalisation (figure 2). Quelques dizaines d'images peuvent suffire pour capturer un style, un vocabulaire formel ou une ambiance. En combinant plusieurs LoRA (architecture vernaculaire, matériaux, styles, etc.) et en pondérant

leur influence, on peut générer des « chimères » qui pourraient nourrir l’imaginaire en phase conceptuelle. Mais même avec des efforts de personnalisation, la pertinence des modèles de diffusion pour l’idéation reste à démontrer en architecture.



Figure 2 : Trois images générées avec SD avec un prompt identique (SD seul, SD+checkpoint d’architecture, SD+checkpoint d’architecture+LoRA Architecturestudio).

Pour les LLM, deux approches dominent pour la personnalisation : le RAG (retrieval-augmented generation), qui connecte un modèle à une base documentaire afin de produire des réponses ancrées dans des sources internes (Lewis et al., 2020), et le fine-tuning, plus complexe, nécessitant plus de traitement de la donnée. Ces techniques, accessibles via la programmation visuelle ou le vibe-coding, permettent de créer des agents conversationnels experts notamment sur des sujets techniques pointus — conception hospitalière, sécurité incendie, réglementation environnementale, rénovation thermique. C’est ce que nous expérimentons actuellement chez Architecturestudio suite au développement d’un chatbot interne.

Enfin, même personnalisés, ces modèles restent hérités d’entraînements initiaux sur des jeux de données massifs que l’on ne maîtrise pas entièrement. La personnalisation ne fait donc pas disparaître, à elle seule, les questions de droit d’auteur, de traçabilité des données, ni les enjeux de responsabilité liés aux contenus générés.

3. L’entraînement de ses propres IA avec ses propres données

Au-delà de l’utilisation ou de la personnalisation de modèles existants, il est possible d’entraîner ses propres IA dès lors que l’on dispose de données — collectées, ou générées (notamment par des méthodes de design génératif). Avec la programmation assistée par IA (qui requiert néanmoins davantage de connaissances informatiques que le seul “vibe coding”), cette démarche devient progressivement plus accessible. Nous présentons ici deux exemples issus de nos travaux de recherche : les modèles de substitution, et la génération d’esquisses fonctionnelles pour des programmes complexes.

L’apprentissage profond peut servir à prédire en temps quasi réel des résultats de simulations physiques (ensoleillement, lumière, thermique, aéraulique) habituellement coûteuses en temps de calcul. Or ces coûts limitent fortement les boucles d’itération entre conception et évaluation. On parle alors de modèles de substitution (Araújo et al., 2023). Pour des

applications contextualisées, des bases de données relativement modestes, comme en Figure 3, peuvent suffire (Duclos-Prévet, 2024).



Figure 3 : Base d'entraînement pour un modèle de substitution pour l'analyse des heures d'insolation annuel à l'échelle d'un quartier urbain

Nous avons ainsi entraîné un modèle de type Pix2Pix (Isola et al., 2017) à prédire différents indicateurs, à plusieurs échelles. Les prédictions se sont révélées très satisfaisantes dès lors que les morphologies testées étaient proches de celles vues pendant l'entraînement (figure 4). Ces résultats ouvrent une perspective opérationnelle : entraîner, au démarrage de chaque projet, un modèle de substitution calibré sur les paramètres réellement pertinents (climat, orientation, variantes de gabarit, etc.). Connecté à l'outil de modélisation, il fournirait un retour quasi en temps réel de l'impact des modifications de conception sur la performance énergétique et le confort.

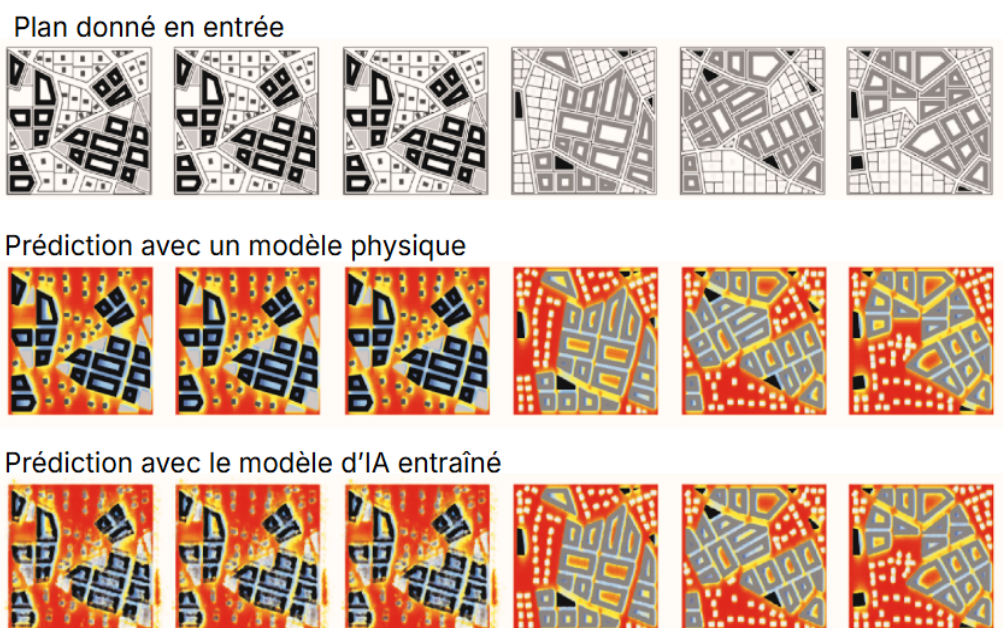


Figure 4 : Test d'un modèle de substitution : morphologies non vues en entraînement (à gauche) et morphologies vues en entraînement (à droite), pour la prédiction des heures d'insolation annuelles.

L'apprentissage profond peut également être mobilisé pour aborder des programmes complexes et fortement contraints par l'usage, comme les équipements hospitaliers. Leur organisation fonctionnelle relève souvent d'un casse-tête combinatoire, où il est parfois impossible pour un cerveau humain de prendre en compte simultanément des centaines de contraintes. Dans ce contexte, optimisation et apprentissage peuvent être combinés pour assister la recherche de solutions. Faute de bases de données suffisamment vastes, nous avons élaboré une méthode de génération procédurale afin de constituer une base synthétique d'entraînement (Figure 5).



Figure 5 : Extrait de la base de données synthétique d'esquisses fonctionnelles pour l'hospitalier.

Notre recherche en cours consiste à croiser apprentissage profond et théorie des graphes (Aalaei et al., 2023; Hu et al., 2020) afin d'entraîner des réseaux capables de proposer des organisations fonctionnelles à partir de données programmatiques. À partir d'une empreinte de bâtiment donnée et d'une liste de services hospitaliers, avec leurs surfaces et contraintes d'adjacence, le modèle apprend à générer plusieurs hypothèses d'implantation cohérentes.

L'IA est un outil puissant et des transformations sont à venir, mais l'architecte n'est pas, selon nous, condamné à les subir. L'enjeu est de choisir l'usage que l'on veut en faire. Nous proposons de mobiliser l'IA comme un levier pour gagner en autonomie et en robustesse sur les dimensions techniques du projet, plutôt que d'automatiser des gestes qui font la valeur et le sens du métier. Une telle posture suppose toutefois de s'intéresser à l'IA au-delà des applications "clé en main", en allant vers des usages plus maîtrisés comme la personnalisation, voire l'entraînement de modèles.

Bibliographie

- Aalaei, M., Saadi, M., Rahbar, M., Ekhlassi, A., 2023. Architectural layout generation using a graph-constrained conditional Generative Adversarial Network (GAN). *Automation in Construction* 155, 105053. <https://doi.org/10.1016/j.autcon.2023.105053>
- Araújo, G.R., Gomes, R., Gomes, M.G., Guedes, M.C., Ferrão, P., 2023. Surrogate Models for Efficient Multi-Objective Optimization of Building Performance. *Energies* 16, 4030. <https://doi.org/10.3390/en16104030>
- Duclos-Prévet, C., 2024. La modélisation des processus d'exploration numérique en Architecture: Design génératif, performance et optimisation. Conservatoire Nationale des Arts et Métiers.
- Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W., 2021. LoRA: Low-Rank Adaptation of Large Language Models. *arXiv e-prints arXiv-2106*.
- Hu, R., Huang, Z., Tang, Y., Van Kaick, O., Zhang, H., Huang, H., 2020. Graph2Plan: learning floorplan generation from layout graphs. *ACM Trans. Graph.* 39. <https://doi.org/10.1145/3386569.3392391>
- Isola, P., Zhu, J.-Y., Zhou, T., Efros, A.A., 2017. Image-to-image translation with conditional adversarial networks, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 1125–1134.
- Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems* 33, 9459–9474.