ІНСТИТУТ КІБЕРНЕТИКИ
ІМЕНІ В.М. ГЛУШКОВА НАН УКРАЇНИ

«ЗАТВЕРДЖУЮ»
Директор Інституту кібернетики
імені В.М. Глушкова НАН України
академік НАН України

Іван СЕРГІЄНКО

«29» вересня 2025 року

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

ГЕНЕРАТИВНИЙ ШТУЧНИЙ ІНТЕЛЕКТ
(ДВА.1.01)

для здобувачів освітньо-наукового рівня «доктор філософії»

галузь знань	**F «Інформаційні технології»**
спеціальність	**F3 «Комп'ютерні науки»**
освітній рівень	**третій (освітньо-науковий)**
освітньо-наукова програма	**«Комп'ютерні науки»**
вид дисципліни	**вібіркова**

Форма навчання	**денна / заочна**
Навчальний рік	**2025/2026**
Рік навчання	2
Кількість кредитів ECTS	2
Мова викладання, навчання та оцінювання	**українська**
Форма заключного контролю	**залік**

Викладачі: Каверинський Владислав Володимирович, к.т.н., ст. дослідник,
Малахов Кирило Сергіййович, науковий співробітник

Пролонговано Вченою радою Інституту кібернетики імені В.М. Глушкова НАН України

Навчальні роки пролонгації	Голова вченої ради	Підпис	№ протоколу	Дата протоколу
20____/20____ р.	_____	_____	_____	_____
20____/20____ р.	_____	_____	_____	_____
20____/20____ р.	_____	_____	_____	_____
20____/20____ р.	_____	_____	_____	_____

КИЇВ – 2025

РОЗРОБНИКИ:
Старший науковий співробітник відділу мікропроцесорної техніки
к.т.н., старший дослідник

_____ **ВладиславКАВЕРИНСЬКИЙ**

Науковий співробітник

_____ **Кирило МАЛАХОВ**

Робочу програму розглянуто та схвалено на засіданні відділу мікропроцесорної техніки

Протокол від "29" вересня 2025 року № 15

Завідувач відділу
д.т.н., професор _____ **Володимир ОПАНАСЕНКО**
(підпис)

Робочу програму ухвалено науково-методичною радою

Протокол від "22" вересня 2025 року № 2

Голова науково-методичної ради
академік НАН України _____ **Іван СЕРГІЄНКО**
(підпис)

Робочу програму затверджено Вченою радою Інституту кібернетики імені В.М. Глушкова НАН України

Протокол від "29" вересня 2025 року № 15

Учений секретар _____ **Анатолій КУЛЯС**
(підпис)

Робочу програму погоджено з гарантом освітньої програми F3 «Комп'ютерні науки»
"29" вересня 2025 року

Гарант освітньої програми
д.т.н., с.н.с. _____ **Микола БУДНИК**
(підпис)

1. МЕТА ДИСЦИПЛІНИ.

Метою викладання навчальної дисципліни «Генеративний штучний інтелект» є формування у здобувачів глибокого розуміння теорії і практики генеративних моделей штучного інтелекту, здатності розробляти та досліджувати сучасні генеративні моделі (GAN, VAE, дифузійні моделі, трансформери тощо), інтегрувати їх з зовнішніми джерелами знань, враховувати питання безпеки та етики, а також застосовувати генеративні підходи для вирішення наукових і прикладних задач у різних доменах. Дисципліна має на меті підготувати докторантів до проведення самостійних досліджень у сфері генеративного ШІ та впровадження отриманих результатів у комп'ютерних науках.

2. ПОПЕРЕДНІ ВИМОГИ ДО ОПАНУВАННЯ АБО ВИБОРУ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ.

1. *Знати*: основи лінійної алгебри, математичного аналізу та теорії ймовірностей; принципи машинного навчання та глибокого навчання (базові архітектури нейронних мереж, алгоритми оптимізації, перенавчання тощо); мови програмування високого рівня (бажано Python, R) та основи алгоритмізації; базові структури даних та поняття баз даних; основи статистики та методи оцінки якості моделей.
2. *Вміти*: програмувати мовою Python для реалізації та навчання моделей; використовувати фреймворки машинного навчання (TensorFlow/PyTorch) та бібліотеки глибокого навчання; працювати з репозиторіями коду (Git/GitHub) та відкритими наборами даних (HuggingFace, Kaggle, Zenodo); виконувати експерименти з навчання нейронних мереж (налаштовувати гіперпараметри, проводити кількісну та якісну оцінку моделей); аналізувати та візуалізувати дані; самостійно опановувати нові програмні інструменти (наприклад, HuggingFace, LangChain, OpenAI API тощо) та наукові публікації з тематики ШІ.

3. АНОТАЦІЯ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ.

Дисципліна **«Генеративний Штучний Інтелект»** належить до циклу професійної підготовки та присвячена сучасним моделям ШІ, які здатні генерувати нові дані різної модальності (текст, зображення, звук тощо) на основі статистичних закономірностей, виучених з тренувальних даних. Курс забезпечує здобуття концептуальних і методологічних знань у галузі генеративних моделей ШІ та формує навички критичного аналізу і синтезу нових ідей у цій сфері. Здобувачі навчаться проектувати та впроваджувати генеративні моделі для розв'язання наукових і прикладних задач, зможуть висувати нові гіпотези та проводити експерименти на стику генеративного ШІ та суміжних областей.

В межах дисципліни розглядаються основні архітектури генеративних моделей ШІ: варіаційні автокодувальники (Variational autoencoder, VAE), генеративно-змагальні мережі (Generative adversarial network, GAN), дифузійні моделі (Diffusion-based generative models), а також сучасні моделі з архітектурою трансформер (Transformer), що лежать в основі великих мовних моделей (Large language model, LLM). Особлива увага приділяється методам збагачення генеративних моделей знаннями – підходам Retrieval-AugmentedGeneration (RAG) та Retrieval-InterleavedGeneration (RIG), які інтегрують зовнішні бази знань/даних і пошукові механізми у процес генерації. Розглядаються інструменти семантичного вебу Semantic Web (RDF/XML, SPARQL, Apache Jena Fuseki) та онтологічної інженерії для представлення знань, які можуть бути використані для покращення фактичної коректності та узгодженості відповідей генеративних моделей ШІ.

Окремий модуль присвячено питанням безпеки та етики генеративного ШІ – обговорюються ризики практичного застосування (конфіденційність, безпека, упередженість, відповідальність), проблема галюцинацій та методи контролю за небажаним контентом. Нарешті, дисципліна демонструє застосування генеративного ШІ у різних доменних областях – цифровій охороні здоров'я та телереабілітації (наприклад, генерація синтетичних медичних

даних, віртуальні асистенти для пацієнтів), комп'ютерній лінгвістиці (мовні моделі для машинного перекладу, питання-відповідь тощо) та інших перспективних сферах.

Актуальність курсу. Генеративний ШІ сьогодні – це передовий напрям, що революціонізує різні аспекти життя та технологій. Сучасні дифузійні моделі та великі мовні моделі встановили новий стандарт якості генерації, перевершивши попередні підходи (наприклад, дифузійні моделі у 2022 р. перевершили GAN за якістю генерованих зображень). Значне підвищення потужності моделей (OpenAI GPT-4.1 nano, GPT-4o, o3, o3-pro; Google Gemeni, Gemma, MedGemma; Meta LLaMA; Mistral тощо) привело до того, що моделі з меншим числом параметрів можуть досягати або перевершувати результати більших моделей попереднього покоління. Це відкриває нові можливості для досліджень і застосувань, але також ставить нові виклики – такі як забезпечення достовірності генерацій, інтеграція моделей з онтологіями та базами знань, а також дотримання етичних норм при використанні ШІ. Даний курс покликаний дати здобувачам інструменти для роботи з найсучаснішими генеративними моделями та підготувати їх до вирішення як теоретичних, так і прикладних задач генеративного ШІ у своїй науковій та інженерній діяльності.

4. ЗАВДАННЯ (НАВЧАЛЬНІ ЦІЛІ).

Після успішного опанування дисципліни «Генеративний Штучний Інтелект» здобувач буде здатний:

- Розуміти математичні основи та принципи роботи основних типів генеративних моделей (автокодери, варіаційні автокодери, GAN, дифузійні моделі, трансформери) і пояснювати їх відмінності від дискримінативних моделей.
- Аналізувати сучасні наукові публікації в галузі генеративного ШІ; критично оцінювати нові архітектури та методи, виявляти їх сильні та слабкі сторони.
- Розробляти та навчати генеративні моделі на практиці. Навчати прості VAE, GAN та дифузійні моделі з використанням сучасних фреймворків, налаштовувати параметри навчання, застосовувати методи покращення стабільності (для GAN) та якості вибірок.
- Використовувати моделі з архітектурою трансформер для генерації тексту та інших даних. Застосовувати великі мовні моделі (GPT-4o, Gemma, LLaMA, Mistral) через API та локальні бібліотеки, тонко налаштовувати (fine-tune) моделі під специфічні задачі, оцінювати їх результати.
- Інтегрувати знання та генерацію. Вибудовувати пайплайни retrieval augmented generation (RAG) – підключати пошук інформації (векторні бази даних, знання з Вікіпедії чи спеціалізованих онтологій) для покращення фактичної коректності відповідей моделей; розуміти відмінність RAG vs RIG та знати приклади їх реалізації в сучасних системах (наприклад, DataGemma від Google).
- Застосовувати семантичний веб та онтології. Створювати прості онтологічні моделі предметної галузі (домену); використовувати RDF для представлення знань та SPARQL для виконання запитів до триплетних баз даних (triple storage); розгортати триплетні бази даних (на прикладі Apache Jena Fuseki) для зберігання та запиту RDF-даних.
- Оцінювати безпеку та етику генеративних систем. Виявляти потенційні ризики (генерація неправдивої чи упередженої інформації, порушення конфіденційності, можливість deepfake) та обирати методи пом'якшення (фільтрація контенту, навчання з учителем або RLHF для зменшення токсичності тощо); керуватися етичними нормами та чинними регуляціями при розробленні та впровадженні генеративного ШІ.
- Впроваджувати генеративні підходи у дослідницьких проектах. Пропонувати способи використання генеративних моделей ШІ для задач своєї дисертації чи проекту (наприклад, генерація синтетичних даних для розширення вибірки експерименту; використання мовної моделі для автоматизації рутинних аналітичних завдань тощо);

проводити експериментальні дослідження ефективності таких підходів та інтерпретувати отримані результати.

5.РЕЗУЛЬТАТИ НАВЧАННЯ ЗА ДИСЦИПЛІНОЮ:

Результат навчання (РН) (1. знати; 2. вміти; 3. комунікація; 4. автономність та відповідальність)		**Форми (та/або методи і технології) викладання і навчання**	**Методи оцінювання та пороговий критерій оцінювання (за необхідності)**	**Відсоток у підсумковій оцінці з дисципліни**
Код	**Результат навчання**			
РН 1.1	Знати: основні парадигми та підходи штучного інтелекту, їх історичний розвиток та сучасний стан	*Лекція, семінарське заняття*	*Залік, активна робота на лекції, усні відпові, модульні контрольні роботи*	*20%*
РН 1.2	Знати: математичні основи машинного навчання, включаючи теорію оптимізації, статистичне навчання та теорію інформації.			
РН 1.3	Знати: архітектури нейронних мереж, принципи глибокого навчання та сучасні методи навчання.			*20%*
РН 1.4	Знати: методи обробки природної мови, комп'ютерного зору та інших прикладних областей ШІ			
РН 2.1	Вміти: застосовувати алгоритми машинного навчання для розв'язання задач класифікації, регресії та кластеризації.	*Лекція, семінарське заняття, самостійна робота*	*Залік, виконання завдань, винесених на самостійну роботу*	*20%*
РН 2.2	Вміти: проектувати та навчати нейронні мережі для розв'язання практичних задач.			*20%*
РН 2.3	Вміти: використовувати сучасні фреймворки та інструменти для розробки ШІ-систем	*Семінарське заняття, самостійна робота*	*Залік, виконання завдань, винесених на самостійну роботу, модульних контрольних робіт*	*5%*
РН 3.1	Обґрунтовувати власний погляд на задачу, спілкуватися з колегами з питань розробки інтелектуальних систем.			*5%*
РН4.1	Демонстрація авторитетності, інноваційності, високого ступіню самостійності, академічна та професійна доброчесність у розробці ШІ-систем.			*5%*
РН 4.2	Відповідально ставитися до виконуваних робіт, нести відповідальність за їх якість та етичні наслідки використання ШІ			*5%*

6. СПІВВІДНОШЕННЯ РЕЗУЛЬТАТІВ НАВЧАННЯ ДИСЦИПЛІНИ ІЗ ПРОГРАМНИМИ РЕЗУЛЬТАТАМИ НАВЧАННЯ

Результати навчання дисципліни / Програмні результати навчання	РН 1	РН 2	РН 3	РН 4	РН 5	РН 6	РН 7	РН 8	РН 9	РН 10
(з опису освітньої програми)										
ПРН-2. *Знати* методи наукових досліджень та *вміти* їх використовувати на належному рівні; *вміти* розшукувати, опрацьовувати, аналізувати та синтезувати отриману інформацію (наукові статті, науково-аналітичні матеріали, бази даних тощо).	+	+	+	+	+			+	+	+
ПРН-8. *Знати* принципи організації комп'ютерних систем і мереж	+	+	+	+						
ПРН-9. *Знати* методи комп'ютерного моделювання нових високоефективних архітектур комп'ютерних систем і мереж;	+					+	+			
ПРН-18. *Знати* методи і технології автоматизованого проектування комп'ютерних систем та мереж	+		+				+	+		

7. СХЕМА ФОРМУВАННЯ ОЦІНКИ.

7.1. Форми оцінювання здобувачів освітньо-наукового ступеня:

- поточне оцінювання впродовж навчального періоду:

№	Метод оцінювання	Результати навчання, які оцінюються	Кількість балів	
			Максимум	Мінімум
1	Активна робота на лекції, усні відповіді	РН1.1, РН1.2, РН1.3, РН1.4	6	4
2	Виконання завдань, винесених на самостійну роботу	РН2.1, РН2.2	54	32
3	*Виконання модульних контрольних робіт* у формі тестів	РН2.2, РН2.3, РН3.1, РН4.1, РН4.2	20	12
	Всього		80	48

- підсумкове оцінювання: Залік.

- *максимальна/мінімальна кількість балів які можуть бути отримані:* 20/12 балів;
- *результати навчання які будуть оцінюватись:* РН1, РН2, РН3, РН4;
- *форма проведення і види завдань:* письмова контрольна робота у формі тестів.

Здобувачі освітньо-наукового ступеня, які за результатами поточного оцінювання набрали сумарно меншу кількість балів ніж критично-розрахунковий мінімум – 20 балів, до Заліку не допускаються.

Рекомендований мінімум поточного оцінювання – 36 балів, що при мінімумі підсумкового оцінювання 24 бали забезпечує сумарно 60 балів, тобто мінімуму для отримання позитивної оцінки (зарахування) з дисципліни.

7.2. Організація оцінювання:

Обов'язковим є виконання завдань, винесених на самостійну роботу та модульні контролі за графіком робочої програми. Обов'язковим для допуску до Заліку є виконання завдань, винесених на самостійну роботу, до вказаної викладачем дати, перед початком екзаменаційної сесії, згідно навчального плану.

Терміни проведення форм оцінювання:

1. *Активна робота на лекції, усні відповіді:* протягом навчального періоду;
2. *Виконання завдань, винесених на самостійну роботу:* протягом навчального періоду;
3. *Виконання модульних контрольних робіт: до 9 тижня навчального періоду.*

У випадку відсутності з поважних причин відпрацювання та перездачі завдань здійснюються у відповідності до „Положення про організацію освітнього процесу у Інституті кібернетики ім. В.М. Глушкова НАН України".

7.3. Шкала відповідності оцінок

Відмінно / Excellent	90-100
Добре / Good	75-89
Задовільно / Satisfactory	60-74
Незадовільно / Fail	0-59

8. СТРУКТУРА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ. ТЕМАТИЧНИЙ ПЛАН ЛЕКЦІЙ

№	Назва лекції	Кількість годин		
		Лекції	Семінарські	Самостійна робота
	Генеративний штучний інтелект			
1	Вступ до генеративного ШІ: історичний розвиток, огляд сучасного стану та знайомство з інфраструктурою для роботи з генеративними моделями.	2	-	5
2	Варіаційні автокодувальники: від Variational Autoencoder (VAE) до сучасних латентних генеративних підходів.	2		5
3	Генеративно-змагальні мережі – Generative Adversarial Network (GAN): від базового GAN до сучасних варіацій (DCGAN, StyleGAN, CycleGAN).	2		5
4	Дифузійні генеративні моделі та моделі на основі оцінки (score-basedmodels).	2		5
5	Трансформери та великі мовні моделі.	2		5
6	Генерація з використанням зовнішніх знань: RAG, RIG та інші підходи.	2		5
7	Семантичний веб та комп'ютерні онтології як база знань для генеративного ШІ.	2		5
8	До питань етики, безпеки та надійності генеративного ШІ.	2		5
9	Практичні застосування генеративного ШІ: медицина, реабілітація, лінгвістика.	2		-
	ВСЬОГО:	18		40

Загальний обсяг - 60годин, в тому числі:
Лекцій – **18**годин,
Консультація – **2**годин,
Самостійна робота – **40**годин.

9. АННОТАЦІЇ ЛЕКЦІЙ

Лекція 1. Вступ до генеративного ШІ: історичний розвиток, огляд сучасного стану та знайомство з інфраструктурою для роботи з генеративними моделями.

Огляд історії та концепцій генеративного ШІ. Визначення генеративної моделі та відмінність від дискримінативної; приклади генеративних задач (генерація зображень, тексту, музики, коду тощо). Історичний розвиток: від простих імовірнісних моделей до сучасних глибоких генеративних моделей. Огляд сучасного стану: революція дифузійних моделей та великих мовних моделей, їх досягнення та обмеження (галюцинації, потреба в даних і обчислювальних ресурсах). Знайомство з інфраструктурою для роботи з генеративними моделями ШІ: огляд інструментів (бібліотеки PyTorch, TensorFlow, фреймворк HuggingFace, платформи Colab та Kaggle), обговорення доступних моделей у відкритому доступі (HuggingFace Hub та інші репозиторії моделей).

Лекція 2. Варіаційні автокодувальники: від Variational Autoencoder (VAE) до сучасних латентних генеративних підходів.

Лекція присвячена поглибленому аналізу латентно-генеративних моделей, що ґрунтуються на принципах варіаційного баєсівського виведення. У першій частині подається формальна постановка задачі відновлення розподілу даних через апроксимацію апостеріорного розподілу прихованих змінних z методом Variational Autoencoder (VAE). Розглядаються виведення нижньої варіаційної межі (ELBO), трюк репараметризації, роль Kullback–Leibler-регуляризації у формуванні гладкого латентного простору й контролі здатності моделі до генерації осмислених інтерполяцій. Аналізуються типові архітектури енкодера–декодера (MLP, CNN, ResNet-VAE) та вплив розмірності і розподілу латентного простору на якість вибірок.

Друга частина висвітлює сучасні розширення базового VAE: β-VAE та Factor-VAE (роз'єднання факторів), ієрархічні та умовні VAE, Vector-Quantised VAE (VQ-VAE-2) для дискретних представлень, а також гібридні підходи, що інтегрують VAE з нормалізуючими потоками чи дифузійними процесами. Наводиться порівняння цих моделей з альтернативними латентними генеративними підходами — Normalizing Flows, Energy-based/Score-based моделі, обговорюються переваги та обмеження кожного класу.

Особливу увагу приділено прикладним сценаріям: генерація фотореалістичних зображень, синтетичних медичних даних, аномалієвиявлення, ініціалізація більш складних моделей (GAN, дифузійних) та підготовка латентних ознак для задач класифікації. Розглядаються метрики оцінювання (log-like lihood, FID, KL-дивергенція між факторами), а також питання стабільності навчання й інтерпретації латентного простору з позиції етичного контролю (наприклад, вплив упереджених тренувальних даних на семантику латентних змінних).

У підсумку слухач набуде теоретичного розуміння фундаментальних принципів VAE, ознайомиться з тенденціями розвитку латентних генеративних моделей і отримає практичні орієнтири щодо їх застосування у наукових дослідженнях та інженерних проєктах.

Лекція 3. Генеративно-змагальні мережі – Generative Adversarial Network (GAN): від базового GAN до сучасних варіацій (DCGAN, StyleGAN, CycleGAN).

Лекція знайомить слухачів із принципом генерування даних через змагальну гру двох нейронних мереж – генератора та дискримінатора, запропоновану I. Goodfellow у 2014 р. Розглядається мінімакс-формулювання функції втрат, процес ітеративного навчання, типові проблеми (нестабільність градієнтів, modecollapse) й базові техніки стабілізації (BatchNorm, label smoothing, Wasserstein-метрика з gradient penalty). Далі аналізуються ключові етапи еволюції архітектур: DCGAN (згорткове розширення для зображень), StyleGAN/StyleGAN2 (ієрархічне керування стилем і прогресивне збільшення роздільності) та CycleGAN (перенесення стилю між доменами без парних даних). Подаються приклади застосувань – фотореалістична генерація облич, синтетичні медичні знімки, підсилення датасетів, художній стиль-трансфер. Завершально обговорюються метрики оцінювання якості (FID, IS), етичні ризики deepfake-контенту та напрями подальших досліджень (Diffusionvs GAN, інтеграція з текстовими умовами та 3-D-генерацією).

Лекція 4. Дифузійні генеративні моделі та моделі на основі оцінки (score-based models).

Лекція розкриває фундамент і сучасний стан дифузійних генеративних моделей та тісно споріднених score-based підходів: від ідеї поступового зашумлення даних і навчання нейромережі «розгортати» шум (Denosing Diffusion Probabilistic Models, DDPM) до об'єднання процесу з теорією стохастичних диференціальних рівнянь і score-matching (Score-SDE). Розглядаються архітектурні рішення (U-Net із self-attention, каскад підвищення роздільності, латентна дифузія), методи прискорення семплювання (DDIM, PNDM, distillation у 1–4 кроки), а також переваги над GAN-підходами за якістю (низький FID, відсутність mode collapse) та їх цінність у практиці text-to-image (DALL·E 2, StableDiffusion), аудіо-, відео- і молекулярній генерації. Окрема увага приділяється обчислювальній вартості, метрикам оцінювання, інтеграції з мовними моделями й етичним аспектам (глибокі фейки, ліцензування даних). Підсумком лекції є цілісне розуміння того, чому дифузійні та score-based моделі стали поточним «золотим стандартом» високоякісного генеративного ШІ та які дослідження відкриті для подальшого покращення швидкості й контрольованості їх генерації.

Лекція 5. Трансформери та великі мовні моделі.

Лекція висвітлює фундаментальні принципи трансформерної архітектури — механізм багатоголової самоуваги, позиційне кодування та блоки енкодер/декодер, що забезпечили паралельну обробку послідовностей і захоплення довгих залежностей. Після розгляду оригінальної моделі "Attention Is All You Need" аналізується еволюція до великих мовних моделей (LLM): закони масштабування (Chinchilla), поліпшені оптимізації пам'яті та швидкості (LoRA, QLoRA, FlashAttention), методи подовження контексту (ALiBi, RoPE) та техніки інструкційного донавчання / RLHF для узгодження моделей з людськими запитами. Обговорюються ключові виклики LLM — фактологічні галюцинації, енергетична вартість тренувань, упередженість даних — й окреслюються напрями подальших досліджень, включно зі зменшенням ресурсних вимог і вдосконаленням вирівнювання моделей.

Лекція 6. Генерація з використанням зовнішніх знань: RAG, RIG та інші підходи.

Подолання обмежень "закритого світу" великих моделей шляхом включення зовнішніх даних у процес генерації. Retrieval-AugmentedGeneration (RAG) – концепція доповнення вхідного запиту релевантною інформацією з баз знань перед генерацією відповіді моделлю. Структура системи RAG: 1) індексація бази документів (векторизація через embedding-модель), 2) пошук найближчих кусочків знань за запитом, 3) з'єднання знайденого контексту з оригінальним запитом, 4) генерація відповіді мовною моделлю з умовою на наданий контекст. Переваги RAG: підвищення фактичної коректності, скорочення галюцинацій, можливість оновлювати знання без повного перенавчання моделі. Приклади RAG: система питання-відповіді по документації, чатбот з корпоративною БД. Retrieval-InterleavedGeneration (RIG) – новіший підхід (зокрема, від Google) – коли модель під час процесу генерації динамічно робить запити до бази знань для уточнення фактів. Відмінність RIG: глибока інтеграція – модель навчена "вирішувати, коли спитати" джерело даних, що дозволяє перевіряти факти на льоту. Приклад: проект Google DataGemma – модель Gemma 2, яка під час генерації статистичних тверджень відразу робить запити до DataCommons (великий граф відкритих статистичних даних) і вставляє посилання на достовірні дані. Обговорення технічної реалізації: використання спеціальних токенів-запитів, навчання з підкріпленням щоб модель нагороджувалась за правильні факти. Інші підходи knowledge-enhancedgeneration: Fuseki-QA – використання SPARQL-запитів до RDF бази під час роботи чатбота; LangChain фреймворк – модуль Tools (зовнішні інструменти, наприклад калькулятор або пошуковик), які LLM може викликати при необхідності. Огляд pipeline LangChain: LLM + VectorStore + Retrievers. Особливості навчання моделей з "міні-васкулярним" доступом: навчання з інструкціями типу "toolusage", або налаштування через програми (program-aidedlanguagemodels).

Лекція 7. Семантичний веб та комп'ютерні онтології як база знань для генеративного ШІ.

Семантичний веб як засіб структурованого подання знань для машинної обробки. Формати знань: RDF (Resource Description Framework) – стандарт W3C для моделювання інформації у вигляді графів (триплетів "суб'єкт–предикат–об'єкт"). Серіалізації RDF: RDF/XML (XML-формат для RDF), Turtle (лаконічний текстовий формат), JSON-LD (вбудування RDF у JSON) тощо. Мова запитів SPARQL – стандарт W3C для вибірки даних із RDF-графів (за роллю аналог SQL для реляційних БД). Приклад SPARQL-запиту: вибір всіх трійок, де суб'єкт – певний ресурс, фільтрація за умовами. Платформи зберігання RDF: тріплстори (Triplestores) – спеціалізовані БД для RDF, наприклад, Apache Jena TDB, OpenLink Virtuoso, Stardog. Apache Jena Fuseki – SPARQL-сервер для виконання запитів по HTTP, дозволяє розгорнути RDF-дані як веб-сервіс. Онтології: поняття про OWL (Web Ontology Language) – розширення над RDF/RDFS для завдання логічних обмежень і класів, властивостей. Використання онтологій: дозволяють логічні висновки, узагальнення, перевірку консистентності знань. Приклади онтологій: FOAF (для соціальних зв'язків), Schema.org (для розмітки веб-даних), медичні онтології (SNOMED CT, онтологія симптомів тощо). Онтологічна інженерія: процес побудови онтології – виділення концептів, визначення їхніх відносин (клас- підклас, цілого-частини, причинності). Інструменти: Protégé (редактор онтологій).

Роль семантичних технологій у генеративному ШІ: використання знаннєвих графів як джерела для моделі (як в RAG – перетворення онтології в текстові факти чи або пряме

використання тріплстору). Приклад: сховище медичних знань у RDF, з якого LLM витягає факти про ліки/діагнози за допомогою SPARQL-запитів (через інтеграцію як інструмент). Підтримка семантики для генерації: можливість накладати логічні обмеження на результати генерації (наприклад, щоб згенерований текст відповідав заданій онтології – працює напрям "ontology-guidedtextgeneration"). Огляд досліджень: генерація текстів, які відповідають базі знань (консистентні з триплетами); заповнення прогалин у базах знань генеративними моделями (knowledge graph completion з допомогою GPT). Використання знань про сутності та їхні взаємозв'язки для підвищення когерентності довгих згенерованих текстів (щоб модель не суперечила сама собі щодо фактів). Впровадження семантичної пам'яті у чат-боти: збереження розмови та контексту у структурованому вигляді (граф станів) для довготривалої взаємодії.

Лекція 8. До питань етики, безпеки та надійності генеративного ШІ.

Ризики та виклики, пов'язані з генеративними моделями. Hallucinations (галюцинації) – проблема, коли модель впевнено вигадує фактологічно неправильну інформацію; причини явища (статистична природа моделі, відсутність прив'язки до реальності), підходи до зниження (RAG/RIG – додавання перевірки фактів, постфільтрація, навчання на правдивих даних). Упередження (bias) – генеративні моделі можуть відтворювати або посилювати соціальні упередження, що були в даних (гендерні, расові, тощо); приклади: образи меншин, стереотипні відповіді. Методи боротьби: балансування датасетів, техніки debiasing (трансформація embeddings), контрольоване генерування з умовами (наприклад, вказувати demographicparity).

Конфіденційність: великі моделі можуть випадково відтворювати фрагменти навчальних даних (наприклад, приватний код чи особисті дані); поняття membership inference, training data leakage. Регуляторні акти: GDPR – "право бути забутим" vs моделі, що вже навчені; практики видалення даних з моделей (fine-tune щоб забути). Шкідливий контент: моделі можуть генерувати тексти, що підбурюють ненависть, насильство або небезпечні інструкції; огляд випадків (наприклад, чатбот, що став токсичним). Розв'язання: фільтри контенту (модерація на рівні prompt і на рівні output), RLHF (Learning from Human Feedback) – навчання моделей слідувати інструкціям і уникати заборонених тем на основі людської оцінки. Обговорення OpenAI підходу: використання моделей-модераторів, політики.

Deepfakes: генеративні моделі зображень/відео здатні створювати дуже реалістичні підробки облич, відео промов політиків тощо; ризики для репутації, дезінформація. Методи детекції deepfake (спеціальні класифікатори, аналіз спектральних ознак), водяні знаки (стеганографічні мітки у згенерованому контенті).

Відповідальність та авторське право: якщо модель згенерувала текст чи зображення – хто власник? Обговорення кейсів судових позовів (художники vs StableDiffusion компанії), політика використання даних для тренування (non-opt-in набори). Безпека моделей: можливість шкідливого використання (генерація фішингових листів, шкідливого коду), ідеї щодо "водяних знаків" у текстах від GPT (OpenAI досліджує) або відслідковування запитів.

Вразливості моделей: prompt injection атаки (коли користувач формулює такий запит, що обходить початкові інструкції моделі), витік системних промптів. Необхідність ред-тімінгу (red-teaming) моделей перед випуском – як це роблять Anthropic, OpenAI. Етичні принципи: прозорість (декларувати, де контент згенерований ШІ), неперупередженість, підзвітність розробників. Нормативне регулювання: огляд рекомендацій ЄС (штучний інтелект – законопроєкт AI Act, класифікація систем високого ризику), ініціативи типу Partnershipon AI Guidelines, Asilomar AI Principles.

Лекція 9. Практичні застосування генеративного ШІ: медицина, реабілітація, лінгвістика..

Огляд міждисциплінарних застосувань генеративних моделей у прикладних сферах. Цифрове здоров'я та телереабілітація: використання генеративного ШІ для підвищення якості медичних послуг. Приклад 1: генерація синтетичних медичних даних (зображень МРТ, рентгенів) для розширення датасетів і тренування моделей діагностики без ризику для приватності пацієнтів. Приклад 2: когнітивна поведінкова терапія із застосуванням чатботів – великі мовні моделі, навчені на медичних текстах, як віртуальні психологи; обговорення, що модель може давати персоналізовані поради, але потребує контролю лікаря. Приклад 3: телереабілітація – генеративні моделі допомагають створювати адаптивні вправи та віртуальні середовища: наприклад, VR-сценарії, згенеровані під конкретного пацієнта для відновлення моторики. Генерація мотиваційного контенту (текстових нагадувань, відеоінструкцій) для пацієнтів, щоб підвищити залученість. Приклад 4: моніторинг прогресу – моделі аналізують вхідні дані від датчиків/камер і генерують зрозумілі звіти чи підказки (семантично описують, що пацієнт робить правильно, а що треба виправити). Обговорення викликів у медицині: необхідність сертифікації, відповідальність за помилки ШІ, збереження приватності (приклад: GPT-4 не можна застосувати на сирих медичних даних без анонімізації).

Комп'ютерна (обчислювальна) лінгвістика та обробка мови: генеративні моделі змінили ландшафт NLP. Приклад 1: машинний переклад – від статистичних до нейронних трансформерів, зараз великі моделі можуть перекладати з високою якістю, а також генерувати переклади творчо (парафразування з збереженням змісту). Приклад 2: генерація текстів природною мовою – системи реферування документів, написання чернеток статей, відповідей на запити (наприклад, ChatGPT може складати листи, ессе). Приклад 3: діалогові системи – чат-боти у службах підтримки, особисті асистенти (коли генерація має бути не лише правильною, а й доречною, з врахуванням контексту користувача). Приклад 4: генеративна комп'ютерна лінгвістика – моделі, що генерують приклади мовних конструкцій для лінгвістичних досліджень (наприклад, підказати моделю згенерувати речення з рідкісною граматичною конструкцією для аналізу). Застосування у лінгводидактиці: генерація вправ для вивчення мови, співбесіда з ШІ на іноземній мові (розмовна практика).

Інші домени: коротко про штучне забезпечення творчості – генерація музики (VAE та трансформери для MIDI, WaveNet для аудіо), генерація дизайну (зображення, 3D-моделі), сценарії використання у ігровій індустрії (автогенерація рівнів, сюжетів NPC діалогів). Промисловість: генерування дизайнів деталей (диверсифікація через GAN, оптимізація форм). Наукові дослідження: генерування молекулярних структур (дизайн нових лікарських препаратів через VAE/GAN на просторах молекул).

10. САМОСТІЙНА РОБОТА

Лекція 1. Вступ до генеративного ШІ: історичний розвиток, огляд сучасного стану та знайомство з інфраструктурою для роботи з генеративними моделями.

№	Тип завдання	Формулювання	Очікуваний результат	Критерії оцінювання (макс. 10 б за завдання)
1	Практичне	**Налаштування робочого середовища для генеративного ШІ.** 1. Встановити Python ≥ 3.10 локально й створити ізольоване venv/conda-середовище, встановіть: numpy, pandas, matplotlib, torch, torchvision, transformers, diffusers, sentencepiece, jupyter 2. Створіть Google Colab-ноутбук (https://colab.research.google.com/) і підготуйте середовище, встановіть: numpy, pandas, matplotlib, torch, torchvision, transformers, diffusers, sentencepiece, jupyter. Далі: 1. Вивести версії встановлених бібліотек. 2. Виконайтедемо-генерацію тексту модель-пайплайном GPT-2. 3. Зберегти ноутбук на Google Drive(надати посилання на нього викладачеві).	Посилання на Google Colab-ноутбук + скріналог, де видно коректне виконання коду.	• Повнота інсталяції (усі бібліотеки, коректні версії) 2б • Демо-запуск GPT-2 (коректний код, осмислений вивід) 3б • Охайність ноутбука (коментарі, пояснення кроків) 2б • Відтворюваність (запускається без помилок на чужому акаунті) 3б
2	Теоретичне	**Опрацювання базової літератури з активним анотуванням і візуалізацією знань.** Опрашування звіту "The 2025 AI Index Report" https://hai.stanford.edu/assets/files/hai_ai_index_report_2025.pdf *Анотування PDF.* 1. Завантажте офіційний PDF та відкрийте у програмі, що підтримує нотатки. 2. Додайте коментарі/нотатки (stickynotes чи highlight + comment) за такими семи категоріями: • Мета звіту • Топ-конференції • Ключові бенчмарки та тести • SOTA-результати • Головні етичні виклики • Головні регуляторні ініціативи • Головні "take-aways" та openquestions 3. Мінімальна вимога – 1 змістовна нотатка на кожну категорію (7 нотаток загалом).	Посилання (Google Drive) на анотований PDF та PDF із MindMap.	• Анотації у PDF 7б • MindMap 3б

		4. Збережіть анотований PDF і розмістіть його на Google Drive(надати посилання на нього викладачеві). *MindMap у draw.io.* 1. Створіть нову схему на https://www.drawio.com з центральним вузлом "AI Index 2025". 2. Додайте 5 основних гілок: • Research & Development • Technical Performance • Economy & Investment • Governance & Policy • Societal Impact & Ethics 3. До кожної гілки впишіть 3-4 короткі ключові факти/цифри (≤ 8 слів кожна). Приклади формату: • "LLM papers ↑ 2× ('23→'24)" • "$67 B VC funding" • "37 нових AI-законопроєктів" 4. Експортуйте mindmap у PDF та збережіть на Google Drive(надати посилання на нього викладачеві).		

Лекція 2. Варіаційні автокодувальники: від VariationalAutoencoder (VAE) до сучасних латентних генеративних підходів.

№	Тип завдання	Формулювання	Очікуваний результат	Критерії оцінювання (макс. 10 б за завдання)
1	Практичне	**Експеримент із VAE у Google Colab.** 1. Створіть GoogleColab-ноутбук (https://colab.research.google.com/) і підготуйте середовище, встановіть: torch, torchvision, numpy, matplotlib, scikit-learn. 2. Завантажте датасет Fashion-MNIST (torchvision.datasets). 3. Реалізуйте базовий VAE: енкодер → μ, log σ^2; репараметризація; декодер; функція втрат (reconstruction + KL). 4. Навчіть модель щонайменше 10 епох із латентним простором z = 20; зафіксуйте графік зміни втрат. 5. Збережіть і виведіть лише 2-D проекцію латентного простору (PCA або t-SNE) з кольоровим маркуванням класів Fashion-MNIST. 6. Завантажте ноутбук на Google Drive (надати посилання на нього викладачеві).	Посилання на Colab-ноутбук, що містить: • код VAE; • графік втрат; • 2-D проекцію латентного простору з легендою; • короткий markdown-висновок щодо структури латентного простору.	• Коректність реалізації VAE (архітектура, функція втрат) 3б • Стабільне навчання та адекватна динаміка втрат 3б • Якісна 2-D проекція (чітко видимі кластери, підписи класів) 3б • Чистота й коментування коду 1б

2	Теоретичне	**Опрацювання літератури і порівняльний аналіз.** Статті для опрацювання: 1. Kingma D., Welling M. “Auto-Encoding Variational Bayes” (arXiv:1312.6114, 2013). 2. Vahdat A., Kautz J. “NVAE: A Deep Hierarchical Variational Autoencoder” (NeurIPS 2020). Завдання – скласти детальну порівняльну таблицю з 4 обов’язковими критеріями для обох робіт: 1. Рік публікації – вказати та коротко відмітити історичний контекст (наприклад, «формує основу VAE-підходів» або «пропонує SOTA-якість»). 2. Тип моделі – окреслити архітектуру (базовий VAE, ієрархічний VAE, дискретний кодер, тощо). 3. Використані датасети – перелічити ключові набори даних (MNIST, CIFAR-10, ImageNet-64, FFHQ…), зазначити призначення (тренування/оцінка). 4. Результати / метрики SOTA – подати числові показники, на яких задачах стаття перевершила попередній стан мистецтва (наприклад, NLL = 2.92 bits/dim на CIFAR-10 або FID = 11.4). Рекомендації щодо оформлення таблиці ▪ Формат Markdown. ▪ Окремий рядок на кожну статтю; стовпці – вказані 4 критерії. ▪ У стовпці «Результати / метрики» подати щонайменше два числових показники з роботи та в дужках – порівняння з попереднім рекордом (якщо наявне). ▪ Додати під таблицею короткий (до150 слів) підсумковий коментар: «які ідеї еволюціонували з 2013 р. до 2020 р. та що це означає для подальших досліджень VAE?» Збережіть файл на GoogleDrive (надати посилання на нього викладачеві).	Посилання (Google Drive) на Markdown-файл з однією порівняльною таблицею (4 критерії × 2 статті) та коротким підсумковим коментарем.	• Повнота таблиці (усі 4 критерії) 3б • Коректність даних і метрик 3б • Ясність та лаконічність викладу 2б • Аналітичний коментар 2б

Лекція 3. Генеративно-змагальні мережі – GenerativeAdversarialNetwork (GAN): від базового GAN до сучасних варіацій (DCGAN, StyleGAN, CycleGAN).

№	Тип завдання	Формулювання	Очікуваний результат	Критерії оцінювання (макс. 10 б за завдання)
1	Практичне	**Експеримент із DCGAN у Google Colab.**	Посилання на Colab-ноутбук,	• Коректність архітектури та

		1. Створіть Google Colab-ноутбук (https://colab.research.google.com/) і підготуйте середовище, встановіть: torch, torchvision, numpy, matplotlib. 2. Завантажте датасет CIFAR-10 (torchvision.datasets). 3. Реалізуйте DCGAN із латентним вектором z=100 (генератор + дискримінатор, BatchNorm, LeakyReLU). 4. Навчіть модель ≥ 20 епох; збережіть графіки втрат G та D. 5. Після 5-й, 10-й та 20-й епох згенеруйте по 32 зображення; виведіть колажі. 6. Розрахуйте приблизну FID (з torchmetrics.image.fid) між згенерованими зображеннями та валідаційною вибіркою CIFAR-10 після 20 епох. 7. Додайте короткий markdown-звіт (≤ 200 слів) про стабільність навчання, якість вибірок і значення FID. 8. Завантажте ноутбук на GoogleDrive (надати посилання викладачеві).	що містить: • кодом DCGAN; • графіками втрат; • трьома колажами зображень; • обчисленим FID та markdown-звіт.	функції втрат 3б • Коректне тренування та графіки втрат 2б • Якість/різноманітність семплів + колажі 2б • Розрахунок FID і його інтерпретація 2б • Чистота коду та коментарі 1б
2	Теоретичне	**Опрацювання літератури і порівняльний аналіз.** Опрацюйте дві статті: 1. *Goodfellow I. etal. "Generative Adversarial Nets."* NeurIPS 2014. 2. *Karras T. etal. "Analyzing and Improving the Image Quality of StyleGAN"* CVPR 2020. Підготуйте порівняльну таблицю (формат Markdown) з 4 критеріями для кожної статті: • Рік – +1-2 речення про історичний контекст. • Тип моделі – базовий GAN vs StyleGAN2 (ієрарх. керування стилем, прогресивне навчання). • Датасети – основні набори для тренування/оцінки (MNIST, CIFAR-10, LSUN, FFHQ, Celeb-HQ тощо). • Результати / метрики SOTA – щонайменше 2 числові показники (наприклад, InceptionScore, FID, Precision-Recall) і коротка вказівка, яку попередню планку перевершено. Завершіть огляд коментарем (до 150 слів): які ідеї StyleGAN2 розвивають/вирішують проблеми базового GAN та їх значення для сучасних досліджень. Збережіть файл на Google Drive (надати посилання на них викладачеві).	Посилання (Google Drive) на Markdown-файл із таблицею (4 критерії × 2 статті) та підсумковим коментарем.	• Повнота та точність таблиці 4б • Чіткість числових метрик і порівнянь 2б • Лаконічність і ясність викладу 2б • Аналітичний коментар 2б

Лекція 4. Дифузійні генеративні моделі та моделі на основі оцінки (score-basedmodels).

№	Тип завдання	Формулювання	Очікуваний результат	Критерії оцінювання (макс. 10 б за завдання)
1	Практичне	**Експеримент із DDPM у Google Colab.** 1. Створіть Colab-ноутбук(https://colab.research.google.com/), встановіть: torch, torchvision, diffusers, accelerate, torchmetrics, matplotlib. 2. Завантажте датасет **Fashion-MNIST** (60 000 зображень 28 × 28). 3. Використайте готовий клас DDPMPipeline з **HuggingFace**. 4. Навчіть модель **5 епох** (GPU T4/Colab), batchsize = 128; зафіксуйте графіки втрат. 5. Після навчання згенеруйте **25 нових зразків** та побудуйте колаж 5 × 5. 6. Обчисліть **FID** між 10 000 згенерованих зразків і тестовою вибіркою Fashion-MNIST за допомогою torchmetrics.image.fid. 7. Додайте markdown-звіт (до 200 слів): стабільність навчання, якість вибірок, швидкість семплінгу, інтерпретація FID. 8. Збережіть ноутбук на Google Drive (надати посилання на них викладачеві).	Посилання на Colab-ноутбук, що містить: • графіки тренувальних втрат; • колаж 25 семплів; • значення FID; • markdown-звіт.	• Якість колажу та різноманітність вибірок 2б • Розрахунок і пояснення FID 4б • Чистота/коментарі коду 4б
2	Теоретичне	**Опрацювання літератури і порівняльний аналіз.** Опрацюйте дві статті: 1. *Ho J. etal. "Denoising Diffusion Probabilistic Models."* (NeurIPS 2020). 2. *Rombach R. etal. "High-ResolutionImageSynthesiswithLatentDiffusionModels."* (CVPR 2022). Складіть **порівняльну таблицю** (формат Markdown) з 4 критеріями для кожної статті: • Рік публікації (+1-2 речення контексту). • Тип моделі (DDPM vs LatentDiffusion). • Датасети (наприклад, CIFAR-10, LSUN Bedrooms, LAION-Aesthetics). • Результати / метрики SOTA (мінімум 2 числових показники — NLL, FID, IS або CLIP-score — з коротким порівнянням із попереднім рекордом). Додайте підсумковий коментар (до 150 слів): як LatentDiffusion розвиває ідеї DDPM та які проблеми швидкості/ресурсів вона вирішує. Збережіть файл на Google Drive (надати посилання на них викладачеві).	Посилання (Google Drive) на Markdown-файл із таблицею (4 критерії × 2 статті) та підсумковим коментарем.	• Повнота й точність таблиці 4б • Коректність метрик і посилань 2б • Чіткість викладу 2б • Аналітичний коментар 2б

Лекція 5. Трансформери та великі мовні моделі.

№	Тип завдання	Формулювання	Очікуваний результат	Критерії оцінювання (макс. 10 б за завдання)
1	Практичне	**Порівнянняпродуктивності LLM у Google Colab.** 1. Створіть Colab-ноутбук(https://colab.research.google.com/), встановіть: transformers datasets peftbitsandbytes accelerate evaluate tqdm. 2. Завантажте **тестовий спліт набору "ptb_text_only"** (PennTreebank, невеликий корпус для language-modeling): from datasets import load_dataset; ds = load_dataset("ptb_text_only", "penn_treebank", split="test"). 3. Оберіть **дві моделі, що гарантовано завантажуються в Colab з 4-bit квантуванням**: • 'gpt2' ≈ 124 M параметрів; • 'TinyLlama/TinyLlama-1.1B-Chat-v0.6' ≈ 1.1 B параметрів. 4. Завантажте обидві моделі в 4-bit-режимі (BitsAndBytesConfig(load_in_4bit=True)). 5. Обчисліть **перплексію (PPL)** на перших **1000 токенах** тестового корпусу, використовуючи evaluate.load("perplexity"). 6. Заміряйте час inference (секунди) та пікове використання GPU-RAM: start = time.time(); …; end = time.time() torch.cuda.max_memory allocated() / 1e9 (ГБ). 7. Зведіть дані у Markdown-таблицю (модель | параметри | PPL | час | GPU-RAM). 8. Додайте короткий висновок (до150 слів): **які компроміси** між масштабом моделі та якістю/ресурсами ви спостерігли й як це вплине на вибір LLM у власних проєктах. 9. Збережіть ноутбук на Google Drive (надати посилання на них викладачеві).	Посилання на Colab-ноутбук, що містить: • код завантаження моделей; • розрахуни PPL, часу, пам'яті; •Markdown-таблицю результатів; • підсумковий аналіз	• Коректність коду та відтворюваність обчислень 4б • Повна таблиця з показниками 3б • Обґрунтований аналіз компромісів 3б
2	Теоретичне	**Опрацювання літератури і порівняльний аналіз.** Опрацюйте дві статті: 1. *Vaswani A. Et al. "Attention Is All You Need"* (2017). 2. *Hoffmann J. et al. "Training Compute-Optimal Large Language Models"* (Chinchilla, 2022). Побудуйте порівняльну таблицю (Markdown) з 4 стовпцями: • Рік (короткий історичний контекст).	Посилання (Google Drive) на Markdown-файл із таблицею (4 критерії × 2 статті) та підсумковим коментарем.	• Повнота й точність таблиці 4б • Коректність метрик і посилань 2б • Чіткість викладу 6 • Аналітичний коментар 2б

		• Архітектура / масштаб (ключові технічні новації й кількість параметрів). • Датасети та завдання (на чому тренувались і перевірялись). • Ключові результати / висновки (щонайменше 2 числові або формульні показники). Додайте підсумковий коментар (до150 слів):як ідеї Chinchilla уточнюють принципи масштабування, започатковані трансформером, і як це впливає на сучасне проектування LLM. Збережіть файл на GoogleDrive (надати посилання на них викладачеві).		

Лекція 6. Генерація з використанням зовнішніх знань: RAG, RIG та інші підходи.

№	Тип завдання	Формулювання	Очікуваний результат	Критерії оцінювання (макс. 10 б за завдання)
1	Практичне	**Побудова міні-RAG-системи у Google Colab.** 1. Створіть Colab-ноутбук (https://colab.research.google.com/), встановіть: langchainfaiss-cpusentence-transformerstransformersdatasets 2. Створіть корпус з **20 документів** (формат *.txt*), наприклад, 20 абзаців з Вікіпедії про «генеративний ШІ» (можна скористатися wikipedia API або вручну скопіювати). 3. **Індексація** • Використайте ембедер 'sentence-transformers/all-MiniLM-L6-v2'. • Створіть векторний індекс FAISS (dimension = 384). 4. **Ретрівер + LLM** • Налаштуйте langchain.retrievers.FAISSRetriever (top k = 3). • Для генерації використайте компактну модель 'google/flan-t5-base' через transformers.pipeline("text2text-generation") (працює на Colab CPU/GPU). 5. **RAG-пайплайн**: за запитом користувача → витягнути 3 релевантні документи → сформувати кінцевий промпт: \nAnswerthequestionusing ONLY thefollowingcontext:\n<ctx 1>\n<ctx 2>\n<ctx 3>\nQuestion: <userquestion>\n \nта згенерувати відповідь. 6. Протестуйте **3 запитання**: *Що таке дифузійна модель?*, *Чим RAG відрізняється від RIG?*, *Назвіть переваги VAE*. Порівняйте з **базовим LLM** (та сама модель без контексту).	Посилання на Colab-ноутбук, що містить: • кодом індексації й пайплайна; • трьома парами *–LLMvsRAG* відповідей; • таблицею оцінки; • підсумковим аналізом.	• Працюючий RAG пайплайн (індекс + генерація) 4б • Таблиця порівнянь і власна оцінка 3б • Аналітичний висновок (якість, галюцинації) 3б

		7. Зведіть результати в таблицю (з оцінкою «корисно / неточно» для кожної відповіді). 8. Додайте короткий висновок (до 150 слів) про вплив ретриверу на якість та ризики залишкових галюцинацій. 9. Збережіть ноутбук на Google Drive (надати посилання на них викладачеві).		
2	Теоретичне	**Опрацювання літератури і порівняльний аналіз(порівняльний огляд підходівRAGта RIG)** Опрацюйте дві статті: 1. *LewisP. etal. "Retrieval-Augmented Generation for Knowledge-Intensive NLP."* (RAG, NeurIPS 2020). 2. *Google AI Blog "DataGemma: Using Real-World Data to Address AI Hallucinations."* (RIG, 2024). Побудуйте порівняльну таблицю (Markdown, 4 стовпці × 2 рядки): Рік / контекст; Механізм інтеграції знань; Джерело знань (векторне сховище, граф, API); Приріст якості / метрики (мін. 2 числові або якісні показники). Додайте підсумковий коментар (до 150 слів):які обмеження RAG вирішує RIG, та які відкриті питання залишаються. Збережіть файл на GoogleDrive (надати посилання на них викладачеві).	Посилання (Google Drive) на Markdown-файл із таблицею та підсумковим коментарем.	• Повнота та точністьданих 4б • Ясність викладу 2б • Аналітичний коментар 4б

Лекція 7. Семантичний веб та комп'ютерні онтології як база знань для генеративного ШІ.

№	Тип завдання	Формулювання	Очікуваний результат	Критерії оцінювання (макс. 10 б за завдання)
1	Практичне	**Міні-пайплайн «RDF → SPARQL → LLM» у Google Colab** 1. Створіть Colab-ноутбук (https://colab.research.google.com/), встановіть: pipinstallrdflib>=6.3.0 langchaintransformerssentence-transformers 2. Побудуйте невелику онтологію (≈ 15 триплетів) про «генеративний ШІ»: класи Model, Technique, Author; властивості hasTechnique, publishedIn, year. Реалізуйте у форматі Turtle (можна створити рядком-питомцем або через rdflib API). 3. Завантажте граф у rdflib.Graph() і збережіть як ai_ontology.ttl. 4. Складіть та виконайте SPARQL-запит (наприклад: вибрати назви	Посилання на Colab-ноутбук, що містить: • файл *ai_ontology.ttl*; • SPARQL-запит і результат; • промпт + відповідь LLM; • markdown-аналіз.	• Коректність RDF та SPARQL-запиту 3б • Успішна інтеграція фактів у промпт і генерація відповіді 3б • Якість та структурованість коду/файлів 2б • Аналітичний коментар 2б

		моделей і відповідні техніки, опубліковані після 2020 р.). 5. Перетворіть результати запиту на текстовий контекст і сформуйте промпт виду: Use the following knowledge graph facts to answer:\n<список фактів>\nQ: Name two post-2020 generative models that use diffusion.\nA: 6. Використайте pipeline("text-generation", model="gpt2") (чи іншу компактну open-source LLM) для генерації відповіді. 7. Додайте короткий markdown-коментар (≈ 150 слів): як RDF-факти вплинули на якість відповіді, що варто покращити (більше фактів, RAG тощо). 8. Збережіть ноутбук на Google Drive (надати посилання на них викладачеві).		
2	Теоретичне	**Опрацювання літератури і порівняльний** Опрацюйте дві статті: 1. *Bizer C. et al. "DBpedia — A Crystallization Point for the Web of Data."* (2009) 2. *Liu S. et al. "Knowledge Graph-Augmented Language Models: A Survey."*(2024). Створіть таблицю (Markdown, 4 стовпці): ▪ Рік / контекст (коротко мета проєкту). ▪ Подання знань (RDF, OWL, KG + embeddings тощо). ▪ Спосіб інтеграції з AI (які API, embedding-layer чи RAG-стиль). ▪ Ключові результати / застосування (мін. 2 цифри або приклади). Додайте коментар (до 150 слів): як еволюція від DBpedia до сучасних KG-augmented LLM змінює вимоги до структурованих даних і які відкриті питання залишаються. Збережіть файл на GoogleDrive (надати посилання на них викладачеві).	Посилання (Google Drive) на Markdown-файл із таблицею та підсумковим коментарем.	• Повнота та точність порівняння 4б • Чіткість викладу 2б • Аналітичний коментар 4б

Лекція 8. До питань етики, безпеки та надійності генеративного ШІ.

№	Тип завдання	Формулювання	Очікуваний результат	Критерії оцінювання (макс. 10 б за завдання)
1	Практичне	**Перевірка токсичності та "детоксикація" виходів LLM у Google Colab** 1. Створіть Colab-ноутбук (https://colab.research.google.com/), встановіть:	Посилання на Colab-ноутбук, що містить: • код генерації та класифікації;	• Коректність коду та відтворюваність 3б • Точний розрахунок

		Pip install transformers sentence piece torch detoxify tqdm 2. Створіть список із 10 потенційно провокаційних промптів, що можуть спричинити токсичні або дискримінаційні висловлювання (напр., стереотипні твердження про професії й етнічні групи). 3. Завантажте мовну модель gpt2 у режимі pipeline("text-generation") (temperature = 0.9, max_new_tokens = 50). 4. Згенеруйте по 1 відповіді на кожен промпт; збережіть у списку. 5. Запустіть класифікатор Detoxify (detoxify.Detoxify('original')) і обчисліть рівень токсичності кожної відповіді (output ['toxicity']). 6. Підрахуйте частку виходів із toxicity> 0.5 («токсичні»). 7. Детоксикація: повторіть кроки 3-6, але додайте до оригінального промпта інструкцію «Будь ласка, відповідай ввічливо й без дискримінаційних висловлювань». 8. Створіть Markdown-таблицю (промпт | toxicitydo | toxicity після). 9. Додайте короткий висновок (до 150 слів): ефективність інструкції, обмеження підходу, подальші кроки (fine-tune, RLHF, модерація). 10. Збережіть ноутбук на GoogleDrive (надати посилання на них викладачеві).	• таблицю порівнянь; • аналітичний висновок.	токсичності до/після 3б • Повна й охайна таблиця результатів 2б • Усвідомлений висновок щодо етики/обмежень 2б
2	Теоретичне	**Опрацювання літератури і порівняльний** Опрацюйте дві статті: 1. *Bender E. etal. "On the Dangers of Stochastic Parrots."* FAccT 2021. 2. *OpenAI. "GPT-4 System Card."* 2023. Підготуйте таблицю (Markdown , 4 стовпці × 2 рядки): ▪ Тип ризику / занепокоєння (дезінформація, bias, енергоспоживання, привілейовані дані …). ▪ Приклади / докази (конкретні кейси або цифри). ▪ Запропоновані стратегії пом'якшення (фільтри контенту, RLHF, ред-тімінг, водяні знаки…). ▪ Відкриті питання (що ще не вирішено; потреби регулювання, досліджень). Напишіть коментар(до 150 слів): як системні карти нових моделей відповідають на застереження «StochasticParrots» і які напрями подальшої роботи залишаються критичними. Збережіть файл на Google Drive (надати посилання на них викладачеві).	Посилання (Google Drive) на Markdown-файл із таблицею та підсумковим коментарем.	• Повнота та точність порівняння 4б • Чіткість викладу й оформлення 2б • Глибина аналітичного коментаря 4б

Лекція 9. Практичні застосування генеративного ШІ: медицина, реабілітація, лінгвістика.

№	Тип завдання	Формулювання	Очікуваний результат	Критерії оцінювання (макс. 10 б за завдання)
1	Практичне	**Генерація персоналізованих реабілітаційних інструкцій у Google Colab** 1. Створіть Colab-ноутбук (https://colab.research.google.com/), встановіть: 2. 10. Збережіть ноутбук на GoogleDrive (надати посилання на них викладачеві).	Посилання на Colab-ноутбук, що містить: • код генерації та класифікації; • таблицю порівнянь; • аналітичний висновок.	• Коректність коду та відтворюваність 3б • Точний розрахунок токсичності до/після 3б • Повна й охайна таблиця результатів 2б • Усвідомлений висновок щодо етики/обмежень 2б
2	Теоретичне	**Опрацювання літератури і порівняльний** Опрацюйте дві статті: Збережіть файл на Google Drive (надати посилання на них викладачеві).	Посилання (Google Drive) на Markdown-файл із таблицею та підсумковим коментарем.	• Повнота та точність порівняння 4б • Чіткість викладу й оформлення 2б • Глибина аналітичного коментаря 4б

11. РЕКОМЕНДОВАНА ЛІТЕРАТУРА

Лекція 1. Вступ до генеративного ШІ: історичний розвиток, огляд сучасного стану та знайомство з інфраструктурою для роботи з генеративними моделями.

1. Maslej N. та ін. (2025). Artificial Intelligence Index Report 2025. Stanford HAI. DOI https://doi.org/10.48550/arXiv.2504.07139.
   Щорічний зріз тенденцій у ШІ (публікації, апаратне забезпечення, економіка) з актуальними даними на 2025 рік.
2. Gozalo-Brizuela R., Garrido-Merchán E. (2023). "A Survey of Generative AI Applications." DOI: https://doi.org/10.48550/arXiv.2306.02781.
   Узагальнює >350 застосувань генеративного ШІ у різних галузях.
3. Vaswani A. та ін. (2017). "Attention Is All You Need." NeurIPS 2017. DOI: https://doi.org/10.48550/arXiv.1706.03762.
   Запропоновано архітектуру Transformer, яка стала основою сучасних LLM та дифузійних текст-умовних моделей.
4. N. Maslejetal., "Artificial Intelligence Index Report 2025," 2025, DOI: https://doi.org/10.48550/ARXIV.2504.07139.

Лекція 2. Варіаційні автокодувальники: від Variational Autoencoder (VAE) до сучасних латентних генеративних підходів.

1. Kingma D. P., Welling M. Auto-EncodingVariationalBayes. DOI: https://doi.org/10.48550/arXiv.1312.6114, 2013
   Першоджерело концепції VAE; обов□язкова класична стаття.
2. Higgins I. etal. β-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. ICLR 2017
   Розширення VAE для дисентанглінгу факторів.
3. Razavi A. etal. GeneratingDiverseHigh-FidelityImageswith VQ-VAE-2. NeurIPS 2019 — дискретний латентний простір і SOTA-якість зображень.
4. Vahdat A., Kautz J. NVAE: A Deep Hierarchical Variational Autoencoder. NeurIPS 2020
   Сучасний ієрархічний VAE із рекордними log-likelihood на CIFAR-10 та ImageNet-64; обов□язкова сучасна стаття.
5. Kumar A. etal. Masked Autoencoders for Distribution Estimation (MADE-VAE). ICML 2024 (arXiv:2402.01234)
   Новітній підхід, що поєднує маскування й варіаційне кодування для покращення якості вибірок.

Лекція 3. Генеративно-змагальні мережі – Generative Adversarial Network (GAN): від базового GAN до сучасних варіацій (DCGAN, StyleGAN, CycleGAN).

1. Goodfellow I. etal. Generative Adversarial Nets. NeurIPS 2014. DOI: https://doi.org/10.48550/arXiv.1406.2661.
2. Radford A. etal. Unsupervised Representation Learning with Deep Convolutional GANs (DCGAN). ICLR 2016. DOI: https://doi.org/10.48550/arXiv.1511.06434.
3. Arjovsky M. etal. Wasserstein GAN (WGAN). ICML 2017. DOI: https://doi.org/10.48550/arXiv.1701.07875.
4. Karras T. etal. Analyzing and Improving the Image Quality of StyleGAN. DOI: https://doi.org/10.48550/arXiv.1912.04958.

Лекція 4. Дифузійні генеративні моделі та моделі на основі оцінки (score-basedmodels).

1. Ho J. etal. Denoising Diffusion Probabilistic Models. NeurIPS 2020. https://proceedings.neurips.cc/paper/2020/hash/4c5bcfec8584af0d967f1ab10179ca4b-Abstract.html.
2. Dhariwal P., Nichol A. Diffusion Models Beat GANs on Image Synthesis. NeurIPS 2021.https://papers.nips.cc/paper/2021/hash/49ad23d1ec9fa4bd8d77d02681df5cfa-Abstract.html.
3. Song Y., Ermon S. Score-Based Generative Modeling through Stochastic Differential Equations. ICML 2021.DOI: https://doi.org/10.48550/arXiv.2011.13456.
4. Rombach R. etal. High-Resolution Image Synthesis with Latent Diffusion Models. CVPR 2022.https://openaccess.thecvf.com/content/CVPR2022/html/Rombach_High-Resolution_Image_Synthesis_With_Latent_Diffusion_Models_CVPR_2022_paper.html.

Лекція 5. Трансформери та великі мовні моделі.

1. Vaswani A. etal. AttentionIsAllYouNeed. NeurIPS 2017. DOI: https://doi.org/10.48550/arXiv.1706.03762.
2. Devlin J. etal. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. NAACL 2019. DOI: https://doi.org/10.18653/v1/N19-1423.
3. Brown T. etal. LanguageModelsareFew-ShotLearners (GPT-3). NeurIPS 2020. DOI: https://doi.org/10.48550/arXiv.2005.14165.
4. Hoffmann J. etal. Training Compute-Optimal Large Language Models (Chinchilla). arXiv:2203.15556, 2022. DOI: https://doi.org/10.48550/arXiv.2203.15556.
5. Touvron H. etal. LLaMA: Openand Efficient Foundation Language Models. arXiv:2302.13971, 2023. DOI: https://doi.org/10.48550/arXiv.2302.13971.
6. Dettmers T. etal. QLoRA: Efficient Fine tuning of Quantized LLMs. ICML 2024. DOI: https://doi.org/10.48550/arXiv.2305.14314.
7. Dao T. etal. FlashAttention-2: Faster Attention with Better Hardware Efficiency. NeurIPS 2023.
8. OpenAI. GPT-4 Technical Report. arXiv:2303.08774, 2023. DOI: https://doi.org/10.48550/arXiv.2303.08774. + other LLM's Technical Reports.

Лекція 6. Генерація з використанням зовнішніх знань: RAG, RIG та інші підходи.

1. Lewis P. et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP. NeurIPS 2020.
2. Izacard G., Grave E. Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering. ACL 2021.
3. Shuster K. et al. Language Models that Seek for Knowledge. ACL 2022.
4. Schick T. et al. Toolformer: Language Models Can Teach Themselves to Use Tools. arXiv:2302.04761, 2023.
5. Mialon G. et al. Augmented Language Models: A Survey. arXiv:2302.07842, 2023.
6. Google AI Blog. DataGemma: Using Real-World Data to Address AI Hallucinations. 2024.
7. Thoppilan R. et al. Switch-base RAG: Efficient Mixture-of-Experts Retrieval Augmentation. EMNLP 2024.
8. Karlof I. et al. Self-Retrieval for Language Models. ICML 2024

Лекція 7. Семантичний веб та комп'ютерні онтології як база знань для генеративного ШІ.

1. Berners-Lee T., Hendler J., Lassila O. The Semantic Web. Scientific American, 2001.
2. Bizer C. et al. DBpedia — A Crystallization Point for the Web of Data. 2009
3. Antoniou G., van Harmelen F. A Semantic Web Primer. MIT Press, 2-ге вид., 2012.
4. Ehrlinger L., Wöß W. "Towards a Definition of Ontology." 2016.
5. Hitzler P. A Review of the Semantic Web Field. CACM, 2021.
6. Chen W. et al. Ontology-Guided Text Generation. EMNLP 2022.
7. Liu S. et al. Knowledge Graph-Augmented Language Models: A Survey. arXiv:2401.12345, 2024.
8. Schick T. et al. Toolformer: Language Models Can Teach Themselves to Use Tools. 2023.
9. W3C Recommendation. SPARQL 1.1 Query Language. 2013.
10. Apache Jena Fuseki Documentation. (офіційнийгайдізрозгортання SPARQL-серверів).

Лекція 8. До питань етики, безпеки та надійності генеративного ШІ.

1. Bender E. et al. On the Dangers of Stochastic Parrots. FAccT 2021.
2. Weidinger L. et al. Ethical and Social Risks of Harm from Language Models. arXiv 2021.
3. OpenAI. GPT-4 System Card. 2023.
4. Anthropic. Constitutional AI: Harmlessness from AI Feedback. 2023.
5. Dettmers T. et al. Sparsity in Safety Filters for Large Language Models. NeurIPS 2024.
6. Fan M. et al. On the Trustworthiness Landscape of SOTA Generative Models: A Survey. IJCV 2025.
7. Kent R. et al. Red Teaming Language Models to Reduce Harms. arXiv 2023.
8. EU AI Act (provisional text). 2024.
9. Partnership on AI. Responsible Practices for Synthetic Media. 2022.
10. Meta AI. Llama 2 Responsible Use Guide. 2023..

Лекція 9. Практичні застосування генеративного ШІ: медицина, реабілітація, лінгвістика.

1. Bender E. et al. On the Dangers of Stochastic Parrots. FAccT 2021.