ChatGPT

Difüzyon Tabanlı Dil Modelleri (dLLM'ler) ve LLaDA: Kapsamlı İnceleme

1. Mimari ve Yapısal Farklar

Transformers ve Model Yapısı: Difüzyon tabanlı dil modelleri (dLLM'ler) mimari olarak genellikle geleneksel büyük dil modellerine benzer şekilde Transformer tabanlıdır. Örneğin LLaDA, standart bir Transformer mimarisini kullanır ancak **dikkat maskesi** konusunda kritik bir fark vardır: LLaDA ve benzeri dLLM'ler *causal mask* (nedensel maskeleme) kullanmaz, dolayısıyla model her konumdaki tüm girdi bağlamını görebilir [1]. Buna karşın geleneksel autoregressive (kendinden regresif) LLM'ler (GPT türevleri gibi) gelecekteki token'ları görmemek için üçgensel (sola dayalı) maskeleme uygular. Sonuç olarak, dLLM'ler metni soldan sağa okuma zorunluluğu olmadan **iki yönlü bağımlılıkları** modelleyebilir; Transformer mimarisi aynı kalsa da olasılık modelleme yaklaşımı farklıdır [2]. Bu sayede dLLM'ler metin içindeki herhangi bir konumun bağlamını tam olarak değerlendirerek, doldurulması gereken maskelenmiş yerleri tahmin eder.

Öğrenme Süreci (Pre-training) Farklılıkları: Geleneksel büyük dil modelleri maksimum olabilirlik ilkesine göre *bir sonraki token'ı tahmin etme* göreviyle eğitilirler. Model, her adımdaki önceki tüm kelimeleri giriş alarak sıradaki kelimeyi üretmeye çalışır. Bu autoregressive paradigma, olasılık dağılımını zincir kuralıyla faktörize eder: $P(x_1, x_2, ..., x_n) = \prod_{i=1}^{n} P(x_i \mid x_{<i})$ [3]. Difüzyon temelli modellerde ise eğitim süreci, *metni bozan ve tekrar oluşturan* iki aşamalı bir süreç etrafında kurgulanır. **İleri süreçte**, orijinal metin belirli bir oranda maskelenerek (gürültü eklenerek) bozulur; **geri süreçte** model bu maskeleri kaldırarak orijinali yeniden oluşturmaya çalışır. LLaDA'nın eğitiminde her eğitim örneğinde maskelenen token oranı rastgele 0 ile 1 arasında seçilir ve model bu rastgele oranda maskelenmiş metindeki gizli token'ları tahmin etmeye çalışır [4]. Bu sayede, bazen çok az kelime maskelenir (kolay görev), bazen neredeyse tüm kelimeler maskelenir (zorlu görev); model her ikisini de görerek hem koşullu tamamlama hem de *model dağılımından örnekleme* yeteneği kazanır. LLaDA'nın eğitim hedefi, model dağılımının olasılıklarını **likelihood bound** (olasılık alt sınırı) optimize edecek şekilde ayarlamaktır [5]. Başka bir deyişle, difüzyon modelleri doğrudan doğruya maksimum olabilirlik hedefine yaklaşmak için *varyasyonel bir hedef* kullanır. Bu yöntem, sabit bir mask oranıyla eğitilen BERT gibi standart maskeli dil modellerinden farklıdır; LLaDA'nın rastgele oranlı maskeleme yaklaşımı, eğitim kaybını modelin negatif log-olabilirliğinin bir üst sınırı haline getirerek tam bir generatif model eğitilmesini sağlar [6] [7]. Böylece dLLM, autoregressive LLM'ler gibi metin olasılık dağılımını öğrenirken, sıradaki kelimeye odaklanmak yerine her adımda eksik parçaları tamamlama şeklinde öğrenir.

Örnekleme Stratejileri: dLLM'lerin en belirgin yapısal farkı, **metin üretme (örnekleme) stratejilerinde** ortaya çıkar. Geleneksel bir LLM, cümleyi soldan sağa doğru birer birer token üreterek tamamlar; her yeni token, önceki tüm token'ları giriş olarak alır. Buna karşın difüzyon tabanlı LLM'ler, tüm çıktıyı bir seferde üretmez, onun yerine çok adımlı bir iyileştirme süreci izler. Örneğin LLaDA'da üretim, başlangıçta tümü maskelenmiş bir dizgiyle başlar ve model her **diffusion adımında** o an maskeli olan tüm konumlar için tahminler yapar [8]. Aşağıdaki görsel, difüzyon tabanlı bir modelin metni **eşzamanlı** olarak nasıl ürettiğini göstermektedir – model ilk başta tüm kelimeleri boş bırakır ve bir taslak yanıtı bir bütün olarak doldurmaya başlar, ardından gerekli yerleri silip yeniden yazarak metni giderek iyileştirir [9]. Bu süreç, sanki metnin önce kaba bir taslağı yazılıyor ve ardından bir editör gibi tüm metin üzerinde düzeltmeler yapılıyormuşçasına işler.

Şekil 1: Autoregressive bir LLM soldan sağa kelime eklerken, difüzyon tabanlı LLaDA rastgele sırayla tüm çıktıyı kademeli olarak inşa eder. Yukarıdaki görsel, "Explain what artificial intelligence is" (Yapay zekânın ne olduğunu açıkla) istemine karşı LLaDA modelinin cevabını oluşturma sürecini göstermektedir. İlk başta tüm kelimeler maskelenmiştir; model çok adımlı bir süreçle her adımda bazı kelimeleri doldurup bazılarını yeniden maskeler ve nihayet tutarlı bir paragraf elde edilir [10]. Bu sayede model, önceki adımlarda ürettiği bir metin parçasını gerekirse sonraki adımlarda düzelterek global tutarlılığı sağlayabilir.

Görüldüğü gibi difüzyon yaklaşımında kelimeler metnin geneline yayılmış şekilde ve eşzamanlı belirir, model gerektiğinde *önceden "söylediği" bir kelimeyi silebilir veya değiştirebilir* [9]. Oysa geleneksel bir autoregressive LLM bir kelimeyi ürettikten sonra geri dönüp onu düzeltme imkânına sahip değildir; çıktıyı baştan sona tek geçişte üretir. Bu yapısal fark, difüzyon modellerine **geri dönüşlü düzenleme** esnekliği kazandırır. LLaDA'nın örnekleme algoritması her adımda maskeli token'ların tümünü tahmin edip düşük güvenli bulduklarının bir kısmını tekrar maskelemeyi içerir; böylece ileri (bozma) ve geri (inşa etme) süreçleri teorik olarak tutarlı hale getirilir [11] [12]. Ayrıca LLaDA, örnekleme sırasında isteğe bağlı olarak **düşük güven skorlu kelimeleri yeniden maskeleme** veya **metni bloklara ayırıp sırayla doldurma** gibi stratejiler de kullanabilir (ör. önce cümlenin ilk yarısını, sonra ikinci yarısını tamamlamak gibi) [12]. Bu stratejiler, üretim kalitesini artırmak veya hızlandırmak için geliştirilmiştir.

Özetle, dLLM'lerin mimarisi Transformer tabanını korurken, *veriyi modelleme ve üretme şekli* farklılık gösterir. Autoregressive LLM'ler tek yönlü (soldan sağa) ve geri alınamaz bir üretim yapısına sahipken, difüzyon tabanlı LLM'ler çok adımlı, iki yönlü bağlam kullanan ve gerektiğinde önceki çıktıları yeniden ele alabilen bir yapıya sahiptir. Bu farklılık, özellikle uzun metinlerde tutarlılık ve esneklik konusunda avantajlar sağlar (aşağıda tartışılacaktır).

2. Matematiksel Farklar: Difüzyon vs. Autoregressive Modelleme

Olasılık Modelleme Yaklaşımları: Autoregressive LLM'ler ve difüzyon tabanlı LLM'ler arasında temel matematiksel fark, metin olasılıklarını modelleme biçiminde yatar. Autoregressive modeller, dil olasılık dağılımını zincirleme kural ile açık biçimde faktörize eder. Bir dizgi $X = (x_1, x_2, ..., x_n)$ için ortak olasılık, belirli bir sıraya göre çarpım şeklinde yazılır:

$$P(X) = P(x_1) \cdot P(x_2 \mid x_1) \cdot P(x_3 \mid x_1, x_2) \cdots P(x_n \mid x_1, ..., x_{n-1}).$$

Bu formülasyon gereği, model her adımda bir sonraki kelimenin koşullu olasılığını öğrenir. Eğitimde çapraz-entropi kaybı ile maksimum olabilirlik (MLE) sağlanmaya çalışılır; yani model, gerçek verideki sıradaki kelimeyi olabildiğince yüksek ihtimalle tahmin etmeye zorlanır.

Difüzyon tabanlı modellerde ise doğrudan bir sıradaki token tahmini yerine, **aynı anda tüm dizginin olasılığını modelleme** yaklaşımı benimsenir. Bu modeller, bir çeşit *gizli değişkenli model* olarak düşünülebilir. Matematiksel olarak, difüzyon süreci genellikle iki yönlü tanımlanır:

- **İleri (forward) difüzyon süreci:** Temiz veri X belirli bir zaman planına göre artan miktarda gürültü ile bozulur. Süreç $X_0 = X$ ile başlar ve X_T tamamen gürültü (ya da maskelenmiş) veriye ulaşana dek adım adım ilerler. Metin için *gürültü*, genellikle token'ların yerini alan özel bir `[MASK]` sembolü veya rastgele semboller olarak gerçekleştirilir. LLaDA özelinde, ileri süreç zaman parametresi $t \in [0, 1]$ aralığında bir orana denk gelir; örneğin $t = 1.0$ tüm token'ların maskelendiği, $t = 0.0$ hiçbir token'ın maskelenmediği durumu temsil eder [13]. Eğitimde her bir cümle için rastgele bir t seçilir ve her token bağımsız olarak bu orana göre maskelenir [4]. Böylece X_t bozulmuş (maskeli) veri elde edilir.

- **Geri (reverse) difüzyon süreci:** Model, bozulmuş veriden orijinale ulaşmayı amaçlar. LLaDA gibi bir model, bir **mask tahminleyici** f_θ (parametreleri θ olan Transformer) kullanarak, X_t içindeki maskelenmiş token'ların orijinal değerlerinin dağılımını tahmin eder [14]. Bu işlem, t değerinin kademeli olarak azaltılmasıyla tekrarlanır; model $t = 1$ (tamamen maskeli) halinden başlayarak $t = 0$ (hiç maskesiz, orijinal metin) haline doğru ilerlerken her adımda biraz daha fazla token'ı doğru doldurmaya çalışır. Geri süreçte her adım, bir önceki adımın çıktısını kısmen koruyup kalanını yeniden maskeler; bu, sürekli bir denoising (gürültü giderme) prosedürü gibi düşünülebilir.

Matematiksel olarak difüzyon tabanlı dil modelleri, eğitim sırasında model parametrelerini ayarlamak için bir **varyasyonel olabilirlik sınırı (ELBO)** optimize eder. LLaDA'nın eğitim kaybı, modelin negatif log-olabilirliğine bir üst sınır oluşturan bir biçimde türetilmiştir [6]. Bu, teknik olarak şu anlama gelir: Autoregressive modellemede doğrudan $\log P(X)$ maksimize edilirken, difüzyon modellemede $\log P(X)$ yerine onun alt sınırını maksimize eden bir kayıp fonksiyonu kullanılır (farklı t adımlarındaki tahmin hatalarının ağırlıklı toplamı gibi düşünülebilir). Bu yaklaşım, difüzyon modelinin de prensipte maksimum olabilirlik öğrenimi yaptığı anlamına gelir – nitekim LLaDA makalesinde, kullanılan kaybın model olasılığının (negatif) logaritmasına üst sınır teşkil ettiği ispatlanmıştır [6]. Tim Kellogg'un ifadeleriyle, LLaDA çalışması “dil modellemede esas yükü çeken şeyin maksimum olabilirlik ilkesi (MLE) olduğunu ve kelimeler arasındaki olasılık ilişkilerini modelleyen herhangi bir yöntemin benzer yeteneklere ulaşabileceğini göstermiştir” [15]. Yani autoregressive paradigma MLE’nin bir uygulamasıyken, difüzyon paradigma özünde aynı ilkeleri farklı bir yoldan gerçekleştirmektedir.

Zincir Kuralı vs. Eşzamanlı Tahmin: Autoregressive modellerde zincir kuralı nedeniyle metin üretimi için bir dizilim (doğal dilde genellikle soldan sağa) seçmek zorunludur. Difüzyon modelleri ise *herhangi bir sıralamaya bağlı değildir*. Matematiksel olarak, LLaDA gibi maskeli difüzyon modelleri *her adımda tüm token’ların tahminini bir dağılımdan çekerek* ortak olasılık dağılımını yaklaşıklar. Bu, RADD adı verilen teorik çalışmada gösterildiği üzere, *herhangi bir sıralamada* autoregressive modelleme yapmanın matematiksel eşdeğerini sağlar [16]. Bir başka deyişle, difüzyon modeli yeterince eğitim aldığında, her adımda maskelenmiş token’ları tahmin etme görevi, sıranın rastgele olması durumunda bile, geleneksel bir dil modelinin herhangi bir sıradaki tahmin görevine denk gelebilir. Bu sayede difüzyon modelleri *ters sıralı metin üretimi* gibi normalde autoregressive modellerin yapmakta zorlandığı görevleri içsel olarak başarabilir hale gelir. Nitekim, LLaDA'nın eğitim hedefinin, *her sıra düzeneğindeki* autoregressive hedefle eşdeğer olduğu ve bu sayede **tersine çevrilmiş diziler (reversal)** konusunda modelin yetkinlik kazandığı teorik olarak gösterilmiştir [17] [18].

Özet: Teknik detaya girmeden toparlamak gerekirse, autoregressive LLM'ler *doğrudan ardıl olasılıkları* öğrenirken, difüzyon LLM'ler *bozma/geri alma süreçleriyle dolaylı olarak* aynı dağılımı öğrenir. Autoregressive modelin matematiği, sıralı koşullu olasılıklar çarpımına dayanır; difüzyon modelin matematiği ise bir dizi yardımcı değişken (maskelenme oranı gibi) üzerinden orijinal veri olasılığını elde etmeye dayanır. Her iki yaklaşım da nihayetinde dil dağılımını yaklaşıklar; LLaDA örneği, difüzyon yoluyla da büyük dil modeli becerilerinin ortaya çıkabildiğini göstererek "*LLM zekâsının kaynağının autoregressive mekanizma değil, özünde generatif modelleme ilkeleri olduğunu*" savunmuştur [19] [14].

3. Modern LLM'lerle Karşılaştırmalı Analiz

Difüzyon tabanlı LLM’ler henüz autoregressive akranlarına kıyasla yeni bir paradigma olmasına karşın, 2025 itibarıyla yapılan çalışmalar umut verici sonuçlar ortaya koymuştur. Bu bölümde doğruluk, üretim kalitesi, hız, bellek ve maliyet gibi açılardan dLLM'lerin (özellikle LLaDA modelinin) modern autoregressive LLM'lerle karşılaştırması sunulmaktadır.

Doğruluk ve Görev Performansı: LLaDA gibi dLLM'ler, kapsamlı karşılaştırmalı değerlendirmelerde birçok görevde geleneksel LLM'lerle boy ölçüşebilmektedir. Nie ve arkadaşlarının 2025 tarihli çalışmasında sunulan LLaDA-8B modeli, kendi verileriyle eğitilmiş benzer boyuttaki autoregressive modellerle (ARM) aynı veri üzerinde test edildiğinde benzer bir ölçeklenme eğrisi sergilemiştir [20]. Örneğin, genel bilgi ve muhakeme gerektiren MMLU testinde LLaDA-8B yaklaşık %65.9 doğruluk elde ederek aynı parametre mertebesindeki autoregressive LLaMA3-8B modeline çok yakın bir sonuç almıştır (%65.4) [21]. Benzer şekilde, sıfır veya birkaç örnekli öğrenme (zero/few-shot) gerektiren 15 farklı görev ortalamasında LLaDA-8B, LLaMA2-7B'yi belirgin farkla geride bırakmış ve LLaMA3-8B ile başa baş performans göstermiştir. Bu bulgular, dLLM'lerin *in-context learning* (bağlamsal öğrenme) yeteneğine de sahip olduğunu ortaya koyar. Özellikle LLaDA, *ölçek büyüdükçe performansını artırma* konusunda autoregressive modeller kadar başarılı görünmektedir – çalışma, büyük veri ve model boyutlarında dLLM'nin tutarlı bir yükseliş sergilediğini göstermiştir [20]. İlginç bir nokta, LLaDA-8B'nin sadece 2.3 trilyon token ile eğitilmiş olmasına rağmen, 15 trilyon token üzerinde eğitilmiş LLaMA3-8B'ye yakın sonuçlar alabilmesidir [21]. Bu da difüzyon yaklaşımının veri verimliliği konusunda potansiyel bir avantajına işaret ediyor olabilir.

Görev özelinde baktığımızda, difüzyon modellerinin güçlü ve zayıf yönleri daha net ortaya çıkar:

- **Mantıksal ve Matematiksel Görevler:** LLaDA özellikle matematiksel muhakeme ve mantık gerektiren görevlerde çarpıcı şekilde iyi sonuçlar vermiştir. Matematik problemlerinden oluşan GSM8K testinde LLaDA-8B, LLaMA3-8B'yi belirgin farkla geçerek %70.7 başarı sağlamıştır; benzer boyuttaki autoregressive model ise %53.1'de kalmıştır [22]. Bir diğer matematik değerlendirmesi olan MATH benchmark'ında da LLaDA (~%27.3), autoregressive modele (~%15.1) üstünlük kurmuştur [22]. Bu görevler genellikle birden çok adım çıkarım ve *tersine akıl yürütme* (verilen sonucu geriye doğru izleme) gerektirebildiğinden, LLaDA'nın çift yönlü bakış açısı avantaj sağlamış olabilir. Nitekim **reversal curse** (tersine çevirme laneti) olarak bilinen ve autoregressive LLM'lerin zayıf kaldığı bir problemi LLaDA başarılı bir şekilde aşmıştır: Ters sırada sunulan bir şiiri tamamlama görevinde LLaDA, GPT-4 benzeri güçlü bir modele karşı bile üstün performans göstermiştir [23]. GPT-4'ün açık sürümü (GPT-4o) ile yapılan bu karşılaştırmada, ileri yönde metin üretiminde GPT-4o daha iyi olsa da, *tersten verilen dizeyi devam ettirme* görevinde LLaDA çok daha yüksek bir başarı yakalamıştır [23]. Bu, difüzyon modellerinin metin yapısına tek yönden bakmadığı için *ters ve alışılmadık sıralı görevlerde* bariz avantajı olabileceğine dair önemli bir göstergedir.

- **Dünya Bilgisi ve Akıl Yürütme:** Genel bilgi, dil anlayışı ve muhakeme gerektiren görevlerde (MMLU, Big-Bench Hard (BBH), ARC, Hellaswag, vb.) LLaDA performansı karışıktır ancak genel olarak aynı jenerasyon AR modellerle rekabet içindedir. Örneğin LLaDA-8B, Winograd tarzı anlama testlerinde (WinoGrande) ~%74.8 doğruluk ile LLaMA3-8B'ye (~%77.3) yakın bir sonuç almıştır [24]. Hellaswag ve PIQA gibi günlük akıl yürütme ve sezgisel fizik testlerinde ise LLaDA (%72–74 civarı) benzer boyut AR modellerden bir miktar düşük kalmıştır (LLaMA3-8B Hellaswag %79, PIQA %80) [24]. Bu farkın olası nedeni, bu tür görevlerde büyük dil modellerinin geniş veriyle öğrendiği dünya bilgisinin rolü olabilir; LLaDA'nın eğitimi daha az veriyle yapıldığından bu alanlarda ufak bir açık gözlenmiştir. Yine de LLaDA'nın sonuçları, Mistral 7B veya LLaMA2 7B gibi önceki nesil modelleri rahatça geçecek düzeydedir [21]. Özetle, difüzyon modeli *genel dil görevlerinde* tutarlı bir şekilde güçlüdür ve en gelişmiş autoregressive modellerin seviyesine yakınsar.

- **Kod Üretimi:** Kod üretimi ve anlama görevlerinde difüzyon tabanlı LLM'ler hem avantaj hem dezavantajlara sahiptir. LLaDA-8B, popüler HumanEval programlama testinde %33.5 başarı oranı yakalayarak LLaMA3-8B ile neredeyse aynı düzeye ulaşmıştır (LLaMA3-8B %34.2) [25]. Bu, temel kod yazma yeteneklerinde en azından benzer ölçekteki bir AR model kadar iyi olabildiğini

gösterir. Öte yandan, kod üzerine özel olarak optimize edilmiş en iyi modeller kadar yüksek başarıya ulaştığını söylemek zordur – örneğin 2024 sonlarında duyurulan Qwen-2.5 (7B) gibi bir model HumanEval'da %57-58 gibi çok daha yüksek bir skor elde edebilmiştir [26]. Yani dLLM'lerin genel kod yazma başarımı, veri ve ince ayar miktarına da bağlı olarak, en iyi AR modellerinin biraz gerisinde kalabilir. **Ancak**, difüzyon modellerinin kod alanında öne çıktığı benzersiz bir durum vardır: *Kodun ortasını tamamlama (fill-in-the-middle)* yeteneği. HumanEval-FIM adlı, ortası boş bırakılmış kodu tamamlama testinde LLaDA-8B ~%73.8 başarı göstermiştir; aynı boyuttaki AR model (LLaMA2-7B) sadece %26.9'da kalmıştır [27]. Hatta LLaDA bu ölçütte LLaMA3 gibi daha yeni bir AR modeli bile hafifçe geçmiştir (LLaMA3-8B %73.3) [27]. Bu sonuç, difüzyon modelinin **herhangi bir konumdaki boşlukları doldurma** kabiliyetinin doğal bir getirisi olarak yorumlanabilir. Autoregressive modeller, ortada boşluk doldurmak için ya özel eğitime ihtiyaç duyarlar ya da sorunu özel bir prompt ile baştan yazmak zorundadır. LLaDA ise maskeli eğitim stratejisi sayesinde, kodun başı ve sonu verilmişken ortasını üretme işini doğrudan gerçekleştirebilmektedir. Bu yönüyle dLLM'ler, kod tamamlama ve düzenleme araçlarında pratik avantajlar sunabilir.

Üretim Kalitesi ve Tutarlılık: Bir modelin sadece doğru cevaplar vermesi değil, aynı zamanda yanıtlarının tutarlı, akıcı ve hatalardan arınmış olması da önemlidir. Difüzyon tabanlı yaklaşımlar bu bağlamda bazı farklılıklar gösterir:

- *Global Tutarlılık:* dLLM'lerin çıktıları tüm cümle boyunca **tutarlılık** açısından avantajlı olabilir. Çünkü metnin her kısmı, üretim sürecinin her adımında yeniden gözden geçirilebilmektedir. Tim Kellogg, difüzyon modelinin "*eşzamanlı düşünme*" kabiliyeti sayesinde birden fazla fikri aynı anda geliştirip tutarsızlıkları en aza indirebileceğini belirtmiştir [28] [29]. Nitekim uzun metin veya belgesel içerik üretiminde tüm bölümlerin birbirine uyumu kritik olduğunda, soldan sağa tek geçiş yapan bir model yerine gerektiğinde geri dönüp düzeltme yapabilen bir modelin daha bütünlüklü bir çıktı verebileceği öne sürülmektedir. Örneğin, uzun bir sözleşme metnini ele alalım: Autoregressive model onu parça parça üretirken önceki bölümlerdeki kararları bazen unutabilir veya çelişebilir; difüzyon modeli ise belgenin genel yapısını her adımda "görerek" gerektiğinde ilk kısımları değiştirip son kısımlarla uyumlu hale getirebilir [30]. Bu sayede metin içinde kendi kendini tekrar eden döngüsel hataların (Tim Kellogg'un tabiriyle "*doom loops*") daha az yaşanabileceği iddia edilmektedir [31]. Bununla birlikte, bu avantajların pratik uygulamalarda ne kadar fark yarattığı henüz tam nicel olarak ölçülmüş değildir; konsept olarak var olan bu esneklik, difüzyon modellerine potansiyel bir kalite üstünlüğü alanı sunar.

- *Halüsinasyon ve Hata Birikimi:* Büyük dil modellerinin en bilinen sorunlarından biri **halüsinasyonlar** (asılsız bilgi uydurma) ve bir hatanın büyüyerek tutarsız yanıtlara yol açmasıdır. Autoregressive LLM'ler, çıkarımlarını adım adım inşa ederken, erken bir adımda yapılan bir hata sonraki tüm adımlara yansıyabilir. Model kendi ürettiği hatalı bilgiyi girdi olarak alır ve onu düzeltme şansı olmadan üzerine inşa eder. Bu bazen kullanıcının hataya dikkat çekmesine rağmen modelin ısrarla hatalı bilgisini savunmasına ("gazetecilik" veya "gaslighting" benzeri durumlar) bile yol açabilir [32]. Difüzyon modelleri ise eğitim itibarıyla *her adımda girişin kısmen doğru kısmen eksik olabileceği* senaryolar gördüğünden, önceki çıktıyı tamamen doğru kabul etme eğiliminde değildir. LLaDA, eğitiminde kendi çıktısını "gerçek" olarak değil, doldurulması gereken bir taslak olarak görmeyi öğrendiği için, halüsinasyon kaynaklı bir yanlış bilgiye aşırı güvenip onu devam ettirme olasılığı daha düşüktür denebilir [33]. Tim Kellogg, LLaDA'nın "kendi çıktısına koşulsuz güvenmek yerine, gerçeği yeniden yaratmak üzere eğitildiğini" vurgulayarak bunun halüsinasyon problemini azaltabileceğini öne sürmüştür [32] [34]. Yani bir dLLM, yanlış bir ara bilgi üretse bile sonraki adımlarda bunu düzeltebilir veya yeniden değerlendirebilir; oysa autoregressive modelde bir kez üretildikten sonra o yanlış bilgi modelin kendi girdi geçmişine sabitlenir.

- *Üretim Akıcılığı:* Dil çıktısının akıcılığı ve dilbilgisel doğruluğu açısından dLLM'ler ile AR modeller arasında bariz bir fark bildirilmemiştir. LLaDA gibi büyük ölçekli bir dLLM, sonuçta Transformer dil temsil gücünü kullandığı için, yeterli eğitimle son derece akıcı ve doğal metinler üretebilir. LLaDA-8B modeli, SFT (denetimli ince ayar) sonrasında çok adımlı diyaloglarda veya açıklayıcı metinlerde tutarlı ve kullanıcı talimatlarına uygun çıktılar vermiştir [35]. Üretim örnekleri incelendiğinde (bkz. LLaDA proje sayfasındaki demolar), modelin yanıtları tutarlılık ve dil kalitesi bakımından autoregressive muadilleriyle benzer düzeydedir. Dolayısıyla akıcılık açısından difüzyon yaklaşımı bir dezavantaj oluşturmamıştır.

Hız, Örnekleme Süresi ve Maliyet: Şu an için difüzyon tabanlı modellerin en önemli dezavantajı **üretim maliyeti ve hızı**dır. Autoregressive bir model, bir çıktının her token'ını tek bir ileri geçiş ile üretir (genellikle önceki adımların ara hesaplamalarını *önbelleğe alarak*). Örneğin 50 token uzunluğunda bir yanıt üretmek için GPT-tabanlı bir model yaklaşık 50 adım (her adımda bir token) hesaplama yapar ve mekanizmalar sayesinde her yeni adımın maliyeti düşük tutulur (KV önbellekleme ile önceki hesaplamaları tekrar yapmaz). Buna karşın LLaDA gibi bir difüzyon LLM, 50 tokenlık bir yanıtı **çok-adımlı difüzyon** ile üretir. LLaDA'nın mevcut en iyi performansı, üretilen token sayısı kadar difüzyon adımı kullanmayı gerektiriyor; yani 50 tokenlık bir yanıt için ~50 yineleme yapılması gerekiyor [36]. Üstelik her yinelemede model tüm diziyi yeniden işler (önbelleğe alma kullanılamıyor). Bu nedenle difüzyon modeli, *satır başına birden çok kez hesaplama yaptığı* için otoregresif modele kıyasla yavaş kalmaktadır. LLaDA ekibinin belirttiği gibi, şu an LLaDA'nın örnekleme hızı AR benzerlerinden düşüktür; bunun üç temel sebebi vurgulanır: (1) LLaDA'nın sabit bir bağlam uzunluğuyla örnekleme yapması (örn. maksimum 4096 token her adım işleniyor), (2) KV-cache (önbellek) kullanımının şu an mümkün olmaması, (3) kaliteli çıktı için çoğunlukla "her token için bir adım" stratejisine ihtiyaç duyması [37]. Adım sayısını azaltmak mümkündür ancak bu durumda çıktı kalitesi bir miktar düşer (çalışmada adım sayısını düşürmenin performansa etkisi deneylenmiştir ve belirgin bir kalite düşüşü gözlemlenmiştir [36]).

Bu yavaşlık, **token başına hesaplama maliyetini** oldukça artırır. Örneğin 1000 tokenlık bir metin oluşturmak, AR bir model için (önbellekleme ile) yaklaşık 1000 birimlik bir işlemken, LLaDA için 1000 adım × 1000 token = 1,000,000 birimlik bir işlem gibi düşünülebilir. Daha somut bir ifadeyle, Nie ve ark.'nın önceki bir çalışması, difüzyon tabanlı maskeli dil modellerinin aynı log-likelihood seviyesine ulaşmak için autoregressive modellere kıyasla ~16 kat daha fazla hesaplama yapması gerekebileceğini öne sürmüştü [38]. Ancak bu sonuçlar görecelidir; pratik görev başarımı ile salt likelihood birebir örtüşmeyebilir. Yine de şu an için dLLM'lerin *ham hız/metin üretim throughput* konusunda AR modellere göre verimsiz olduğu söylenebilir.

Bellek Tüketimi: Bellek kullanımı açısından durum daha karmaşıktır. Bir autoregressive model, önbellekleme kullandığında bellekten feragat edip hızı artırır; her yeni token için tüm katmanların anahtar-değer özetlerini saklar, bu da uzun çıktılarda hatırı sayılır hafıza kullanımı demektir. Difüzyon modelinde böyle bir önbellek tutulmadığından, her adım tüm hesaplamayı yeniden yaptığından anlık bellek kullanımı sabittir ve çıktı uzunluğundan bağımsızdır. Bu, *aynı donanım üzerinde* difüzyon modelinin tek seferde daha kısa bağlam kullandığı durumlarda bellek avantajına bile dönüşebilir. Ancak genellikle dLLM'ler pratikte daha yavaş oldukları için aynı işi yapmak üzere daha uzun süre GPU meşgul ederler. LLaDA mimarisinde, KV-cache olmadığı için bazı optimizasyonlar devre dışı kalmıştır; örneğin LLaMA gibi modellerde verimi artıran "grouped-query attention" tekniği LLaDA'da kullanılamamıştır [39]. Bu yüzden LLaDA, aynı parametre sayısını korumak için besleme-ileri katman boyutunu bir miktar küçültmek zorunda kalmıştır [40]. Bu değişiklikler model boyutunu dengede tutmak içindir ve bellek tüketimine dolaylı etkileri olabilir. Özetle, dLLM'lerin **eğitim parametre boyutu** ve **gereken bellek** açısından AR modellerden bir farkı yoktur (LLaDA 8B ve LLaMA3 8B benzer büyüklükte modellerdir). Fark, daha çok **çalıştırma sırasında** ortaya çıkar: AR modeller bellek kullanarak hesaplamayı kısmen önbelleğe alabilirken, dLLM'ler her adıma tam hesaplama yaparlar. Yine de bu, bellekten ziyade zaman maliyetini artıran bir farktır.

Optimizasyon ve Gelecek İyileştirmeler: Difüzyon modellerinin hız sorunları, araştırmacılar tarafından aktif bir çalışma konusudur. Görüntü difüzyon modellerinde 2019-2023 arası dönemde inanılmaz hız iyileştirmeleri gerçekleştiğine dikkat çekilmektedir (örneğin, DDPM'den **Consistency Models**'a kadar yaklaşık 1000 kat hızlanma elde edilmiştir) [41]. Benzer bir trendin dil modellerinde de olması beklenebilir. LLaDA ekibi, örnekleme verimliliğini artırmak için bazı yöntemleri tartışmaktadır: *Yarı-autoregressive örnekleme* (metni bloklara bölerek kısmen sırayla üretme) ve *consistency distillation* (öğrenilmiş bir hızlandırma tekniği) bunlardan bazılarıdır [41] [12]. Consistency distillation, difüzyon adımlarının sayısını azaltmak için modelin farklı adımlarda tutarlı çıktılar verecek şekilde eğitilmesini sağlar; görüntü alanında bu sayede adım sayısı büyük oranda düşürülebilmiştir. Ayrıca difüzyon modellerine *önbellek mantığı* eklemek üzere araştırmalar da vardır (örn. "Fast-dllm" ve "dllm-cache" gibi erken çalışmalardan bahsedilmektedir) [42]. 2025 itibarıyla bu optimizasyonlar tam olgunlaşmamış olsa da, gelecek vaat eden yönlerdir.

Token Başına Maliyet: Maliyeti somutlaştırmak için token başına maliyeti de ele alalım. Bir autoregressive modelde genellikle bir token üretmenin maliyeti 1 forward pass'tir (küçük bir ek maliyetle). Difüzyon modelinde ideal kalite için her token defalarca işlenir. Bu durum, şu an için dLLM'leri bulut tabanlı servislerde çalıştırmayı pahalı hale getirir. Ancak eğitim maliyeti konusunda LLaDA örneği dikkat çekicidir: 8B parametreli bir modelin 2.3T token üzerinde eğitimi ~0.13 milyon GPU-saat (H800) sürmüştür ki, bu rakam benzer ölçekli bir autoregressive LLM eğitimiyle aynı mertebededir [43]. Yani *eğitim maliyeti* olarak difüzyon yaklaşımı rekabetçi olabilir; asıl fark *çıktı üretim maliyetinde* ortaya çıkmaktadır.

4. Kullanım Bağlamlarında dLLM ve LLaDA Performansı

Difüzyon tabanlı dil modellerinin pratikteki faydalarını ve zorluklarını daha iyi anlamak için, onları farklı kullanım alanlarında değerlendirmek gerekir. Aşağıda LLaDA başta olmak üzere dLLM'lerin **doğal dil üretimi**, **kod yazımı** ve **diyalog sistemleri** gibi alanlardaki performansı ve özellikleri ayrı ayrı incelenmiştir.

4.1 Doğal Dil Üretimi (Genel Dil Modeli Kullanımları):

Genel metin üretimi, özetleme, soru-cevap, metin tamamlama gibi görevlerde difüzyon tabanlı LLM'ler, beklenen temel yetenekleri sağlayabilmektedir. LLaDA, geniş bir ön eğitimden ve ardından talimat verilerine süpervizeli ince ayardan geçtiği için, bir kullanıcı istemine anlamlı ve tutarlı doğal dil yanıtlar verebilir. Örneğin LLaDA-Instruct modeli, "Yapay zekâ nedir, açıkla" gibi bir isteğe verdiği yanıtta birkaç cümlede yapay zekâyı tanımlayıp örnekler sunabilmektedir [44] [45]. Bu yanıtlar, akıcılık ve doğruluk açısından LLaDA'yı autoregressive rakiplerinden ayırt etmenin zor olduğunu gösterir.

dLLM'lerin belki de en ilginç yönü, **metin tamamlama stratejilerindeki esneklik**tir. Örneğin bir paragrafın ortasına bir cümle eklemek veya verilen başlangıç ve bitiş cümlelerine uygun bir orta metin üretmek difüzyon modeli için doğal bir görevdir. Bu tür bir kullanım, roman veya senaryo yazımı gibi yaratıcı alanlarda faydalı olabilir; yazarın bazı bölümlerini yazdığı, bazı bölümlerini modelin doldurduğu senaryolar difüzyon modeliyle daha organik çalışabilir. Autoregressive modeller de bu tür görevler için **in-filling** teknikleriyle uyarlanabilir, ancak bu genellikle modele ek pozisyonel işaretçiler verilmesini veya özel eğitim gerektirir. LLaDA ise eğitimden itibaren bu duruma aşina olduğu için, doğal dil üretiminde *esnek tamamlama* yeteneğine sahiptir.

Kalite tarafında, dLLM'lerin halüsinasyon eğilimi tamamen ortadan kalkmasa da potansiyel olarak daha yönetilebilir olabilir. Model, cevabını bir kerede üretmediği, defalarca üzerinden geçtiği için, bariz tutarsızlıkları kendi başına fark edip giderebilir. Bu durum henüz sistematik çalışmalarla kanıtlanmamış olsa da, kullanıcılardan gelen anekdotlar önemlidir. Örneğin, difüzyon modelinin bir paragrafı üretirken,

paragraf sonunda girişle çelişen bir ifade yazdıktan sonra bir sonraki difüzyon adımında bunu düzelttiği senaryolar gözlemlenmiştir (otoregresif modelde ise çelişki, üretildiği anda sabitlenir ve genelde model geriye dönük düzeltme yapamaz). Sonuç olarak, makale yazımı, özet çıkarma veya rapor oluşturma gibi doğal dil üretimi bağlamlarında dLLM'ler en az geleneksel modeller kadar başarılı olabilmekte ve bazı durumlarda **editöryel tutarlılık** avantajıyla daha bile iyi sonuçlar verebilmektedir.

4.2 Kod Yazımı ve Yazılım Geliştirme:
Kod üretimi, LLM'lerin giderek önem kazanan bir kullanım alanıdır. LLaDA gibi difüzyon modelleri, kod yazarken farklı güçlü yanlar sergiler. İlk olarak, bir kod dosyasının herhangi bir yerine kod ekleme/düzeltme yeteneği pratikte çok değerlidir. Yazılımcılar sık sık var olan kodun ortasına yeni satırlar ekler veya bir fonksiyonun gövdesini daha sonra doldururlar. Autoregressive bir model, metnin sadece sonuna ekleme yapabilir; ortasına ekleme yapması için genelde "önceki kısım + [boşluk] + sonraki kısım" şeklinde bir giriş verilerek, boşluğa karşılık gelen çıktının sonra birleştirilmesi gerekir. LLaDA ise doğrudan bu boşluğu tek seferde doldurabilir. Bu, kod tamamlama araçlarında difüzyon modelini çekici hale getirir. Nitekim, HumanEval-FIM testindeki üstün performansına değinmiştik – LLaDA, orta-doldurma senaryosunda küçük AR modellerin ulaşamadığı başarıya ulaşmıştır [27] .

Öte yandan, tam uçtan uca kod generasyonu (örneğin sıfırdan işlevsel bir kod parçası yazma) konusunda dLLM'lerin mevcut durumu biraz muhafazakâr ilerliyor. LLaDA-8B'nin HumanEval sonucu, parametre boyutu benzer bir GPT türeviyle aynı düzeydeydi ancak en iyi özel kod modellerine yetişemedi. Bunun muhtemel nedenlerinden biri, LLaDA'nın eğitim verisinin genel amaçlı oluşu ve kod için özel optimize edilmemiş oluşudur (2.3T token içinde kod verisi bulunsa da, GPT-4 veya specialized Code LLM'lerin gördüğü devasa miktarda GitHub verisiyle kıyaslanamaz). Dolayısıyla, bir yazılım geliştirme yardımcısı olarak LLaDA-Instruct, standart bir 8B LLM'in yapabileceklerini yapar: Fonksiyon taslağı önermek, basit algoritmaları kodlamak, bilinen kütüphane çağrı örneklerini üretmek vb. Ancak çok karmaşık veya uzun kodlar söz konusu olduğunda, parametre boyutu ve veri kısıtı nedeniyle yanıtları yetersiz kalabilir.

Difüzyon modelleriyle kod alanında deneyebilecek bir diğer yenilik, **adım adım kod iyileştirme** olabilir. Örneğin, önce hatalı çalışan bir kod parçasını üretip sonra difüzyon adımlarıyla hataları ayıklamak teorik olarak mümkün. AR modellerin aksine, difüzyon modeli aynı kod üzerinde birden fazla geçiş yaparak semantik hataları düzeltebilir (yeter ki hatayı tespit edebilsin). Bu, gelecekte kod-debug için dLLM kullanımına kapı aralayabilir. Şu an bu konuda spesifik deney raporları bulunmamakla birlikte, dLLM mimarisinin buna elverişli olduğu söylenebilir.

Sonuç olarak, yazılım geliştirme bağlamında dLLM'ler halihazırda **kodu istenen yere ekleme/doldurma** kabiliyetiyle öne çıkıyor. Genel kod yazma kalitesinde en iyi AR modelleri henüz yakalanmış değil, fakat difüzyon yaklaşımının kendine has esneklikleri mevcut. İleride daha büyük difüzyon modelleri veya kod verisiyle özel eğitilmiş sürümler (belki *LLaDA-Code* tarzı modeller) bu alanda rekabeti kızıştırabilir.

4.3 Diyalog ve Konuşma Sistemleri:
Büyük dil modellerinin popüler kullanım alanlarından biri de insanlarla doğal dilde sohbet edebilen **diyalog sistemleri**dir (örn. sanal asistanlar, ChatGPT benzeri sohbet botları). LLaDA, denetimli ince ayar aşamasında özellikle talimatları izleme ve çok döngülü diyalog yürütme konusunda eğitilmiş bir versiyon sunmuştur (LLaDA-Instruct). Bu model, tıpkı ChatGPT veya benzeri bir fine-tuned LLM gibi, kullanıcının çok adımlı diyalog isteklerine yanıt verebilir. Örneğin LLaDA-Instruct ile yapılan bir etkileşimde, modelin çok adımlı bir çeviri diyaloğunu başarıyla yürüttüğü görülmüştür: Kullanıcı önce bir dizeyi Çinceye çevirmesini istemiş, ardından aynı dizenin Almancaya çevirisini istemiş, LLaDA her iki isteği de doğru şekilde yerine getirip soruları sırasıyla yanıtlamıştır [46] [47] . Yine bir başka örnekte, çok aşamalı bir diyalogda kullanıcının sorduğu şiir dizesinin kaynağına dair soruya LLaDA doğru yanıtı

vermiş, ardından kullanıcının yönlendirmesiyle bu yanıtı Çinceye ve Almancaya çevirmiş ve en sonunda da “hayat seçimleri hakkında 5 cümlelik bir şiir” yazma isteğini yerine getirmiştir [48] [49]. Tüm bu etkileşim, LLaDA’nın bağlama uygun kalabildiğini ve *konuşmanın bağlamını sürdürme* konusunda başarılı olduğunu göstermektedir.

Kullanıcı talimatlarını izleme ve uygun üslupta yanıt verme (instruction-following) konusunda LLaDA, SFT sonrasında belirgin bir iyileşme sergilemiştir [35]. Bu, difüzyon yaklaşımının diyalog için bir engel teşkil etmediğini, aksine verildiğinde aynı şekilde öğrenebildiğini ortaya koyar. Hatta, LLaDA-Instruct’ın bazı diyaloglarda LLaMA-2 gibi modellerden daha az çelişkili veya tutarsız olabileceği öne sürülmüştür. Bunun olası nedeni, difüzyon modelinin cevaplarını bir seferde üretmediği için, cümlenin başı ve sonu arasında anlamsal kopuklukları kendi iç mekanizmasıyla düzeltebilmesidir. Örneğin, uzun bir açıklamada başlangıçta “Bu konuda emin değilim” diyen model, sonradan fikrini netleştirip kesin bir yanıt verme kararı alırsa, difüzyon sürecinde ilk tereddüt cümlesini silip yerine daha tutarlı bir giriş yazabilir. Autoregressive model ise bir kere “emin değilim” dediyse, devamında fikrini değiştirse bile metinde o ifade kalmış olur ve cevap bütünlüğüne zarar verebilir.

Diyalog sistemlerinde bir diğer kritik konu **güvenlik ve saçma yanıtların engellenmesi**dir. LLaDA’nın bu açıdan özel bir avantajı olup olmadığı net değil, zira asıl belirleyici olan eğitim verisinin niteliği ve uygulanan güvenlik filtresidir. Ancak, Tim Kellogg’un işaret ettiği gibi difüzyon modeli kendi çıktısına körü körüne inanmadığı için, önceki turda söylediği hatalı bir şeyi düzeltmeye daha açık olabilir [32] [34]. Yine de, kullanıcı ile çok turlu etkileşimlerde her tur bağımsız bir yeni üretim olarak ele alındığından, pratikte difüzyon modeli de her turu bir başlangıç kabul eder. Yani ChatGPT benzeri bir uygulamada, LLaDA’nın her kullanıcı mesajından sonra yeni bir difüzyon süreci başlar; bu süreç içinde global bir tutarlılık sağlasa da, turlar arası tutarlılık AR ve difüzyon modelleri için benzerdir (önceki konuşma geçmişi basitçe yeni tura giriş olarak verilir).

Performans metriklerine dair, LLaDA-Instruct’ın çeşitli diyaloğa yönelik kıyaslamalarda yer aldığı bilinmektedir. Örneğin açık kaynak model değerlendirmelerinde, kullanıcının yönergelerine uygun ve faydalı yanıt verme bakımından LLaDA’nın, benzer boyuttaki diğer ince ayarlı modellerle yarıştığı aktarılmıştır. Ancak GPT-4 gibi devasa ve RLHF (insan geribildirimiyle ince ayarlı) modellerle boy ölçüşmek henüz beklenmemelidir. LLaDA 8B, parametre ve eğitim verisi olarak orta ölçekli bir modeldir; bu nedenle çok karmaşık çok adımlı mantık yürüten sohbetlerde zorlanabilir.

Diyalog bağlamında difüzyon modelinin **ilginç bir artısı**, aynı anda birden fazla yanıt olasılığını düşünsel olarak değerlendirebilmesidir. Örneğin, bir kullanıcının sorusuna yanıt üretirken, model difüzyon adımları boyunca birkaç farklı yaklaşımı metne yansıtıp sonra en mantıklısını bırakacak şekilde kendi kendine bir **iç müzakere** yapabilir. Bu durum, tıpkı bir insanın önce birkaç olasılığı aklından geçirip sonra birini söylemesine benzetilebilir. Autoregressive model ise genelde tek bir akış üretir. Bu spekülatif bir avantaj olsa da, gelecekte *kendi kendine tartışarak cevap bulma* (self-debate) yöntemlerinin difüzyon modellerine entegre edilebileceği düşünülebilir.

Özetlemek gerekirse, diyalog sistemlerinde dLLM'ler: - Kullanıcı mesajlarını anlama ve uygun yanıtlar üretme konusunda başarılıdır (talimat takip yeteneği SFT ile sağlanmıştır). - Çok tur boyunca bağlamı koruyabilir ve tutarlı davranabilir (bunda AR modellerden geri kalmaz). - Difüzyon süreci sayesinde, tek bir yanıt içinde önceki cümlelerini revize etme şansına sahip olduğundan, yanıtın bütünlüğü ve tutarlılığı yüksek olabilir. - Şu an için yanıt hızında AR modellere göre yavaştır, bu nedenle gerçek zamanlı sohbet uygulamalarında optimizasyon gerektirir. Örneğin bir kullanıcı sorusuna cevap verirken LLaDA birkaç saniye gecikmeyle yanıt üretebilir, oysa optimize edilmiş AR modeller daha akıcı cevap verebilir. Bu, mühendislik çözümleriyle aşılabilecek bir konudur (ör. daha az difüzyon adımı ile kabaca yanıt üretip sonra arkada kaliteli hale getirmek gibi yöntemler düşünülebilir).

Sonuç: 2025 itibarıyla difüzyon tabanlı dil modelleri, LLaDA örneğinde görüldüğü gibi, geleneksel büyük dil modellerine **gerçek bir alternatif** oluşturabilecek seviyeye gelmiştir. Mimari ve matematiksel olarak farklı bir yol izleseler de, bu modeller temel dil anlama ve üretme yeteneklerini başarıyla edinmiştir. Autoregressive paradigmaya kıyasla en büyük üstünlükleri, *esnek üretim stratejileri* (herhangi bir sırada üretim, infilling) ve *global tutarlılık/yineleme imkanı*, en büyük zayıflıkları ise *örnekleme hızının düşüklüğü* ve *mevcut optimizasyonların olgunlaşmamış olması* şeklinde özetlenebilir. Akademik çalışmalar LLaDA gibi bir modelin 8 milyar parametre ölçeğinde bile diyalog, programlama, matematik, çeviri gibi çeşitli alanlarda sağlam performans verdiğini göstermiştir [35] [50]. Bu alanların her birinde dLLM'ler kendine has bazı avantajlar sunmaktadır: Örneğin doğal dilde kavramsal bütünlük, kodda kısmi tamamlama, diyalogda kendi hatasını düzeltme gibi. Önümüzdeki dönemde, difüzyon modellerinin verimlilik sorunları aşıldıkça ve ölçekleri büyüdükçe, modern LLM'lerle yarışlarının daha da kızışması beklenmektedir. Şimdiden yapılan çalışma, "*LLM becerilerinin yalnızca autoregressive modellere özgü olmadığını*" ispat etmiş ve literatürde yeni bir araştırma yönü açmıştır [51].

Kaynaklar: Bu incelemede sunulan bilgiler ve karşılaştırmalar, 2024-2025 yıllarında yayımlanmış güncel akademik makalelerden ve teknik raporlardan derlenmiştir. Özellikle LLaDA modelini tanıtan *Large Language Diffusion Models* makalesi ve ekibin teknik notları yoğun biçimde referans alınmıştır [52] [17]. Ayrıca difüzyon dil modelleri üzerine diğer önemli çalışmalar olan *SMDM (Scaling Masked Diffusion Models)*, *MD4 (Masked Diffusion for Discrete Data)* ve *MDLM (Masked Diffusion Language Models)* ilgili bölümlerde dolaylı olarak referans edilmiştir. Tim Kellogg gibi araştırmacıların yorumları ise kavramsal farkları vurgulamak amacıyla kullanılmıştır [53] [15]. Bu alan hızla gelişmekte olup, önümüzdeki yıllarda difüzyon ve autoregressive yaklaşımların hibrit yöntemlerle de bir arada değerlendirilmesi olasıdır – araştırmacılar şimdiden difüzyon modellerine yönelik yeni iyileştirme teknikleri üzerinde çalışmaya başlamışlardır [41]. Bu rapor, 2025 itibarıyla gelinen noktayı kapsamlı bir şekilde özetlemeyi hedeflemektedir.

---

[1] [3] [4] [5] [6] [8] [11] [12] [14] [19] [20] [21] [22] [23] [24] [25] [26] [27] [38] [39] [40] [43] [50] [52] [2502.09992] Large Language Diffusion Models
https://ar5iv.labs.arxiv.org/html/2502.09992v2

[2] [7] [16] [17] [18] [36] [37] [41] [42] GitHub - ML-GSAI/LLaDA: Official PyTorch implementation for "Large Language Diffusion Models"
https://github.com/ML-GSAI/LLaDA

[9] [10] [15] [28] [29] [30] [31] [32] [33] [34] [53] LLaDA: LLMs That Don't Gaslight You - Tim Kellogg
https://timkellogg.me/blog/2025/02/17/diffusion

[13] [44] [45] [46] [47] [48] [49] Large Language Diffusion Models
https://ml-gsai.github.io/LLaDA-demo/

[35] [51] [2502.09992] Large Language Diffusion Models
https://arxiv.org/abs/2502.09992