

**TOBB EKONOMİ VE TEKNOLOJİ ÜNİVERSİTESİ**  
**FEN BİLİMLERİ ENSTİTÜSÜ**

**ÇİZGE SINIR AĞLARI VE DERİN TAKVİYELİ ÖĞRENME KULLANARAK  
OTOMATİK MOLEKÜL ÜRETİMİ**



**YÜKSEK LİSANS TEZİ**

**Rıza IŞIK**

**Bilgisayar Mühendisliği Anabilim Dalı**

**Tez Danışmanı: Doç. Dr. Mehmet TAN**

**NİSAN 2022**



## ÖZET

Yüksek Lisans Tezi

### ÇİZGE SİNİR AĞLARI VE DERİN TAKVİYELİ ÖĞRENME KULLANARAK OTOMATİK MOLEKÜL ÜRETİMİ

Rıza IŞIK

TOBB Ekonomi ve Teknoloji Üniversitesi  
Fen Bilimleri Enstitüsü  
Bilgisayar Mühendisliği Anabilim Dalı

Tez Danışmanı: Doç. Dr. Mehmet TAN

Tarih: Nisan 2022

Belirli istenen özelliklere sahip moleküler yapılar inşa etme fikri, ilaç keşfi ve malzeme bilimindeki birkaç büyük problemin temelini oluşturur. Fakat, kimyasal uzay o kadar geniştir ki, günümüzde mevcut olan hesaplama kaynakları ile bu alanın tamamını araştırmak imkansızdır. Bu nedenle, yeni bir ilaç bulmak hem zaman hem de maddi açıdan çok maliyetli olabilir. Neyse ki, son yıllarda artan veri ile birlikte bilgisayar donanımı ve yazılımındaki gelişmeler, doğal dil işlemeden bilgisayarla görüye uzanan alanları etkileyen derin yapay sinir ağlarında bir devrime yol açmıştır. Derin öğrenme, moleküler özelliklerin tahmini ve optimizasyonu için yeni yolların tasarımı da dahil olmak üzere, ilaç keşfindeki bir dizi alanı etkilemiştir. Derin öğrenmedeki bu gelişmeler ve ilaç üretimi için kullanılacak büyük veri tabanlarının artışı sayesinde son yıllarda umut verici sonuçlar alınmıştır.

Bu tez çalışmasında, istenen belirli özellikler için yeni moleküller üretebilen bir model geliştirilmiştir. Bu model temelde iki farklı modelin birleşiminden oluşmaktadır. İlk model, moleküllerin üretilmesini gerçekleştiren bir Derin Takviyeli Öğrenme mimarisine sahiptir. Diğer model ise Derin Takviyeli Öğrenme mimarisine sahip modelin eğitiminde kullanılan moleküler temsilleri üreten bir Çizge Sinir Ağı'dır. Bu Çizge Sinir Ağı modeli için de iki farklı mimari kullanılmıştır. Bu mimarilerden ilki, Çizge Evrimsel Ağı, diğeri ise Çizge Dikkat Ağı'dır. Geliştirilen modeli kullanarak temelde beş ayrı iş için molekül optimizasyonu ve üretimi gerçekleştirilmiştir. Bu işler, QED değerinin optimizasyonu, benzerlik kısıtlaması ile cezalı LogP değerinin optimizasyonu, benzerlik kısıtlaması ile QED değerinin optimizasyonu, belirlenen hedef protein alıcıları (F2, PPAR, JAK2) için kenetlenme skorunun optimize edilmesi ve kenetlenme skorunun

optimize edildiđi iřteki eđitilen model kullanılarak rastgele bařlangıç moleküllerinin optimize edilmesi řeklinde dir. Önerilen bu modeli kıyaslamak için literatürde benzer görevlerde en iyi sonuçları almıř bir model olan MolDQN, temel yöntem olarak tercih edilmiřtir. Kıyaslamak için kullanılan bu model aynı Derin Takviyeli Öđrenme mimarisine sahiptir ancak eđitilirken molekül temsili olarak Morgan Parmak İzi'ni kullanmaktadır. Yapılan deneylerin sonuçları, Çizge Sinir Ađı mimarisini kullanarak üretilen moleküllerin, Morgan Parmak İzi yöntemi kullanılarak üretilen moleküllerden belirgin düzeyde daha başarılı olduđunu göstermiřtir.

**Anahtar Kelimeler:** Çizge sinir ađları, Derin takviyeli öđrenme, Hesaplamalı molekül tasarımı, Moleküler toksisite



## ABSTRACT

Master of Science

### AUTOMATED MOLECULE GENERATION USING GRAPH NEURAL NETWORKS AND DEEP REINFORCEMENT LEARNING

Rıza IŞIK

TOBB University of Economics and Technology  
Institute of Natural and Applied Sciences  
Department of Computer Engineering

Supervisor: Assoc. Prof. Mehmet TAN

Date: April 2022

The idea of building molecular structures with certain desirable properties underlies several major problems in drug discovery and materials science. However, the chemical space is so vast that it is impossible to explore the entire space with the available computational resources today. Therefore, finding a new drug can be very costly in terms of time and money. Fortunately, advances in computer hardware and software, combined with increasing data in recent years, have led to a revolution in deep neural networks affecting areas ranging from natural language processing to computer vision. Deep learning has influenced several fields in drug discovery, including the design of new pathways for the prediction and optimization of molecular properties. Due to these developments in deep learning and the increase in large databases that can be used for molecule generation, encouraging results have been obtained in recent years.

In this thesis, a model that can generate new molecules for certain desired properties has been developed. This model consists of a combination of two different models. The first model has a Deep Reinforcement Learning architecture that handles the generation of molecules. The other model is a Graph Neural Network that generates molecular representations used in training the Deep Reinforcement Learning architecture. Two different architectures are used for this Graph Neural Network model. The first of these architectures is the Graph Convolutional Network and the other is the Graph Attention Network. By using the proposed model, molecule optimization and generation were carried out for basically five different jobs. These tasks include optimization of the QED value, optimization of the penalized LogP value with similarity constraint, optimization of the QED value with similarity constraint, optimization of the docking score for the

desired target protein receptors (F2, PPAR, JAK2), and optimization of the random starting molecules using the trained model in the previous task for which the docking score is optimized. To compare this proposed model, MolDQN, a model with the best results in similar tasks in the literature, was preferred as the baseline method. This baseline method has the same Deep Reinforcement Learning architecture but uses the Morgan Fingerprint as the molecule representation while being trained. The results of the conducted experiments show that the molecules generated using the Graph Neural Network architecture were significantly more successful than the molecules generated using the Morgan Fingerprint method.

**Keywords:** Computational molecule design, Deep reinforcement learning, Graph neural networks, Molecular toxicity



## TEŐEKKÜR

Yüksek lisans eğitimin ve çalışmalarım boyunca yardım ve desteğini hiçbir zaman benden esirgemeyen sevgili hocam Doç. Dr. Mehmet TAN'a, lisans ve yüksek lisans eğitimim boyunca bana kazandırdıkları deneyim ve tecrübe için TOBB Ekonomi ve Teknoloji Üniversitesi Bilgisayar Mühendisliği bölümünün kıymetli öğretim üyelerine ve yüksek lisans öğrenimim boyunca bana sağladığı çalışma ve burs imkânı için TOBB Ekonomi ve Teknoloji Üniversitesi'ne ve TÜBİTAK'a teşekkürlerimi sunarım.

Bu uzun, zorlayıcı ama bir o kadar da keyifli olan süreçte, beni yalnız bırakmayan ve her zaman destekleyen sevgili aileme ve arkadaşlarıma teşekkür ederim.







## İÇİNDEKİLER

	<u>Sayfa</u>
<b>ÖZET</b> . . . . .	<b>iv</b>
<b>ABSTRACT</b> . . . . .	<b>vi</b>
<b>TEŞEKKÜR</b> . . . . .	<b>viii</b>
<b>İÇİNDEKİLER</b> . . . . .	<b>ix</b>
<b>ŞEKİL LİSTESİ</b> . . . . .	<b>xi</b>
<b>ÇİZELGE LİSTESİ</b> . . . . .	<b>xiii</b>
<b>KISALTMALAR</b> . . . . .	<b>xv</b>
<b>1. GİRİŞ</b> . . . . .	<b>1</b>
<b>2. İLGİLİ ÇALIŞMALAR</b> . . . . .	<b>4</b>
<b>3. ÖN BİLGİ</b> . . . . .	<b>7</b>
3.1 SMILES . . . . .	7
3.2 Morgan Parmak İzi Temsili . . . . .	7
3.3 Takviyeli Öğrenme . . . . .	7
3.4 Q Öğrenme . . . . .	9
3.4.1 Derin Q öğrenme . . . . .	10
<b>4. YÖNTEMLER</b> . . . . .	<b>12</b>
4.1 Derin Takviyeli Öğrenme Modeli (MoldQN) . . . . .	12
4.2 Çizge Sinir Ağı Modelleri . . . . .	14
4.2.1 Çizge evrimsel ağı . . . . .	15
4.2.2 Çizge dikkat ağı . . . . .	15
4.3 Önerilen Model . . . . .	17
<b>5. DENEYSEL SONUÇLAR</b> . . . . .	<b>20</b>
5.1 Model Ayarları . . . . .	21
5.1.1 Derin Q ağı modeli (DQN) . . . . .	21
5.1.2 Çizge tabanlı modeller . . . . .	21
5.2 QED Optimizasyonu . . . . .	22
5.3 Benzerlik Kısıtlaması ile Cezalı LogP Değerinin Optimizasyonu . . . . .	23
5.4 Benzerlik Kısıtlaması ile QED Optimizasyonu . . . . .	25
5.5 Hedef Proteine Kenetlenme Skorunun Optimize Edilmesi . . . . .	27
5.5.1 F2 (protease) kenetlenme . . . . .	28
5.5.2 PPAR (nuclear receptors) kenetlenme . . . . .	29
5.5.3 JAK2 (kinase) kenetlenme . . . . .	31
5.6 Öğrenilmiş Q Fonksiyonu ile Molekül Üretimi . . . . .	31
<b>6. SONUÇ</b> . . . . .	<b>34</b>
6.1 Değerlendirmeler ve Tartışma . . . . .	34
6.2 Gelecek Çalışmalar . . . . .	36
<b>KAYNAKLAR</b> . . . . .	<b>38</b>
<b>ÖZGEÇMİŞ</b> . . . . .	<b>43</b>



## ŞEKİL LİSTESİ

	<u>Sayfa</u>
Şekil 3.1: Takviyeli öğrenmenin genel işleyişi. . . . .	8
Şekil 3.2: Derin Q öğrenmede kullanılan bir derin yapay sinir ağının örneği. . .	11
Şekil 4.1: Morgan Parmak İzi'nin elde edilme süreci. . . . .	17
Şekil 4.2: Çizge modellerinden elde edilen molekül temsilleri. . . . .	18
Şekil 4.3: Önerilen modelin işleyişi. . . . .	19
Şekil 5.1: Son 100 adımda üretilen moleküllerin QED skorlarının dağılımı. . . .	23
Şekil 5.2: 0,6 benzerlik kısıtlaması değeriyle ile cezalı LogP optimizasyonu için eğitilen modellerin ilk 5 adımında oluşan çıktılarının örnekleri (birinci sıra GAT ile, ikinci sıra GCN ile, üçüncü sıra ise ECFP ile eğitilmiştir ve SIM benzerlik skorunu göstermektedir). . . . .	25
Şekil 5.3: ECFP ve MACCS temsillerinin benzerlik skorlarına ilişkin kıyası. . .	26
Şekil 5.4: Son 100 adımda üretilen moleküllerin ödül değerlerinin dağılımı. . .	28
Şekil 5.5: Üretilen moleküllerin ödül değerlerinin dağılımı (F2 için). . . . .	29
Şekil 5.6: Üretilen moleküllerin ödül değerlerinin dağılımı (PPAR için). . . . .	30
Şekil 5.7: Üretilen moleküllerin ödül değerlerinin dağılımı (JAK2 için). . . . .	32
Şekil 5.8: Üretilen moleküllerin toplam ödül dağılımı (F2 için). . . . .	33
Şekil 5.9: Üretilen moleküllerin toplam ödül dağılımı (PPAR için). . . . .	33
Şekil 5.10: Üretilen moleküllerin toplam ödül dağılımı (JAK2 için). . . . .	33



## ÇİZELGE LİSTESİ

	<u>Sayfa</u>
Çizelge 5.1: En iyi üç molekülün QED skorları. . . . .	23
Çizelge 5.2: Ortalama cezalı LogP artışı. . . . .	26
Çizelge 5.3: En iyi ödül değerine sahip moleküllerin QED ve benzerlik skorları. . . . .	28
Çizelge 5.4: En iyi ödül değerine sahip üç molekül (F2). . . . .	29
Çizelge 5.5: En iyi ödül değerine sahip üç molekül (PPAR). . . . .	30
Çizelge 5.6: En iyi ödül değerine sahip üç molekül (JAK2). . . . .	31





## KISALTMALAR

<b>DNN</b>	: Derin Sinir Ađı (Deep Neural Network)
<b>DQN</b>	: Derin Q Ađı (Deep Q Network)
<b>ECFP</b>	: Geniřletilmiř Bađlantı Parmak İzi
<b>GAT</b>	: Çizge Dikkat Ađı (Graph Attention Network)
<b>GCN</b>	: Çizge Evriřimsel Ađı (Graph Convolutional Network)
<b>LogP</b>	: Bir Kimyasalın Oktanol ve Su Arasındaki Bölünme Katsayısının Logaritması
<b>LSTM</b>	: Uzun Kısa Süreli Bellek (Long Short-Term Memory)
<b>MDP</b>	: Markov Karar Süreci (Markov Decision Process)
<b>RL</b>	: Takviyeli Öğrenme (Reinforcement Learning)
<b>RNN</b>	: Özyinelemeli Sinir Ađları (Recurrent Neural Network)
<b>TD</b>	: Zamansal Fark (Temporal Difference)
<b>SMILES</b>	: Basitleřtirilmiř Moleküler Girdi Hattı Giriř Sistemi
<b>SVM</b>	: Destek Vektör Makinesi (Support Vector Machine)
<b>QED</b>	: İlaç Benzerliđinin Kantitatif Tahmini
<b>QSAR</b>	: Kantitatif Yapı Aktivite İliřkisi





## 1. GİRİŞ

Bilgisayar destekli algoritmalar ve yöntemler, kimya bilişiminde özellikle ilaç üretimi ve tasarım yöntemleri ile önemli küçük moleküllerin geliştirilmesinde uzun yıllardır önemli bir rol oynamaktadır. Bu yöntemler temelde iki farklı yaklaşıma dayanır. Bunlardan ilki, yapıya dayalı yöntemlerdir. Bu yapı tabanlı yöntemler, bir molekülün veya bir ligandın yapısal bilgisini hedef olarak kullanarak geliştirilmesini amaçlar. Bu tarz yöntemler prensipte yapı bilgisi mecburi olduğu için yüksek verimli taramaya da benzemektedir. Bir diğer yöntem ise ligand tabanlı olarak sınıflandırılır. Bu liganda dayalı yöntemlerde, literatürde bilinen aktif olarak kullanılan ligandlara benzerliğine bağlı olarak aktivite tahmini için yalnızca ligand bilgileri kullanılır [1].

Sıfırdan veya belirli bir başlangıç noktasından, istenilen özelliklere sahip moleküller tasarlamak çok değişkenli bir optimizasyon görevidir. Ayrıca, organik moleküllerin kimyasal uzayının büyüklüğü göz önünde bulundurulduğunda bu görev çok zorlayıcıdır. Küçük organik moleküllerin kimyasal uzayında yaklaşık olarak  $10^{60}$  kimyasal yapının olduğu tahmin edilmektedir [2]. Bu nedenle, geleneksel yöntemler ile sıfırdan istenilen özelliklerde ilaç (molekül) üretilmesi veya belirli ilaçların optimize edilmesi hem maddi hem de zaman açısından maliyetli olmasının yanı sıra birçok açıdan da riskli bir süreçtir. Son yıllarda artan veri ve hesaplama gücü sayesinde yapay öğrenme yöntemleri ile birçok alanda çok daha başarılı sonuçlar alınabilmektedir. Özellikle, bu yöntemlerden biri olan derin öğrenme algoritmaları, başta akademik çalışmalar olmak üzere birçok sektörde aktif bir şekilde kullanılmaktadır. Kimya ve biyoloji bilişiminde artan veri ile birlikte günden güne gelişmekte olan bu teknoloji, moleküler özelliklerin tahmin edilmesi ve bu özelliklerin optimize edilmesinin yanı sıra, sıfırdan ilaç üretimi için yeni yolların tasarlanmasında ve ilaç keşfindeki birçok noktayı etkilemektedir. Genel olarak bu çalışmalar, geliştirilen yöntemleri kullanarak istenilen özelliklerde sıfırdan molekül üretimi veya mevcut moleküllerin optimizasyonu işlemlerini bilgisayar ortamında otomatikleştirerek zaman ve maddi açıdan masraflı olan bu süreci en aza indirmeyi amaçlar.

Bu tez çalışmasında, [3]'te geliştirilen derin takviyeli öğrenme yapısını (MolDQN), çizge tabanlı modelleri kullanarak moleküllerin temsillerinin elde edildiği başka bir yapıyla birleştirerek oluşturulan yeni bir model tasarlanmıştır. Tasarlanan bu modeli, belirlenen moleküler yapıların özellikleri kullanılarak, girdi olarak verilen başlangıç molekülleri farklı görevlerde optimize edilmiştir. Ek olarak bazı görevlerde, istenilen yapısal özellikler için sıfırdan moleküller de üretilmiştir. Derin takviyeli öğrenme yapısı MolDQN'de olduğu gibi herhangi bir veri kümesine ihtiyaç duyulmadan eğitim işlemi

uygulanabilir. Böylece teorik olarak, yanlı olmayan bir uzayda sıfırdan moleküller üretilebilir veya mevcut moleküller geliştirilebilir. Ayrıca, yine bu tasarımın yapısından dolayı tasarlanan ortam gereği belirli kurallar doğrultusunda moleküler değişiklikler yapıldığı için üretilen her molekül %100 geçerlidir.

MolDQN yapısı gereği derin Q ağını eğitirken molekülleri temsil etmek için Morgan Parmak İzi temsillerini kullanmaktadır. Bu çalışmada MolDQN yapısının molekül temsili üreten kısmı yeniden tasarlanarak çizge tabanlı yapay sinir ağlarından alınan moleküler temsiller kullanılmıştır. Çizge tabanlı modeller ile yapılan çalışmaların da gösterdiği gibi [4], moleküllerin çizge yapısı ile birlikte kullanılması parmak izi temsillerine göre deneysel anlamda daha iyi sonuçlar verebilmektedir. Burada da bunu dikkate alarak, Çizge Evrişimsel Ağı ve Çizge Dikkat Ağı olmak üzere iki farklı çizge sinir ağı mimarisi kullanılmıştır. Dikkat edilmesi gereken bir başka konu da takviyeli öğrenme ile tasarlanan yapay öğrenme modellerinin eğitim sürecidir. Bu modeller yapısı gereği zaman açısından maliyetli olabilmektedir. Bu nedenle, zaman açısından maliyetli olan eğitim sürecini nispeten kısaltmak için önceden eğitilmiş çizge tabanlı modellerin kullanılması tercih edilmiştir. İki farklı çizge tabanlı yapay sinir ağı mimarisi için iki farklı önceden eğitilmiş model kullanılmıştır. Bu iki önceden eğitilmiş model, aynı veri kümesinde (Tox21 Challenge [5]) moleküler toksisite tahmini görevlerinde kullanılmak üzere eğitilmiştir.

Bu çalışmanın iki ana motivasyonu vardır. İlk olarak, takviyeli öğrenme tasarımındaki durum temsiline öneminden dolayı, çizge sinir ağı modellerinden alınan molekül temsillerinin MolDQN yapısı ile birleştirilerek oluşturulan yeni tasarımın etkinliğinin araştırılması amaçlanmıştır. Bahsedildiği üzere, önceden eğitilen modeller toksisite tahmini için eğitilmiştir ve bir kimyasalın oktanol ve su arasındaki bölünme katsayısının logaritmasının (LogP; optimize ettiğimiz hedef özelliklerden biri) bu toksisite ile ilişki olduğu bilinmektedir [6]. Bu nedenle, önceden toksisite tahmini için eğitilmiş modeller ile LogP arasındaki korelasyonun, geliştirilen modeldeki etkisinin incelenmesi, ikinci motivasyonu oluşturmaktadır.

- Bölüm 1’de tez çalışmasının genel olarak amacı ve tasarlanan model özetlenmiştir.
- Bölüm 2’de tez çalışması ile ilgili literatürde bulunan çalışmalar özetlenmiştir.
- Bölüm 3’de tez çalışmasında kullanılan bazı yöntemler hakkında kısa bir ön bilgi paylaşılmıştır. Özellikle buradaki çalışmada bir türevinin tasarlandığı takviyeli öğrenme yönteminin temelde nasıl çalıştığı hakkında bilgi verilmiştir. Ayrıca temel yöntemde kullanılan molekül temsili olan ECFP ve literatürde kimyasal bilgi işleme için kullanılan SMILES yapısı özetlenmiştir.
- Bölüm 4’de belirli hedefler ve moleküler özellikler için sıfırdan molekül üretimi

ve optimizasyonu görevlerinde kullanılan önerilen modelden bahsedilmiştir. MolDQN yapısı ve bu yapıya ilave edilen çizge sinir ağı mimarilerinin detayları aktarılmıştır.

- Bölüm 5’de önerilen model ile yapılan deneylerin sonuçları aktarılmıştır. Bu sonuçlar ile temel yöntemden alınan sonuçların kıyaslandığı çizelgeler ve şekiller raporlanmıştır. Deneylerin nasıl ve hangi ayarlar kullanılarak yapıldığından detaylıca bahsedilmiştir.
- Bölüm 6’da raporlanan sonuçlar tartışılmıştır ve çalışmanın bazı sınırlarından bahsedilmiştir. Bu sınırların gelecek çalışmalarda nasıl çözülebileceği aktarılıp tez sonlandırılmıştır.



## 2. İLGİLİ ÇALIŞMALAR

İlaç üretimi ve keşfi için son yıllarda yapılan çalışmalarda artan veri ile birlikte yapay öğrenme yöntemlerinin de bu alandaki etkisi gün geçtikçe artmaktadır. Molekül optimizasyonu ve bununla birlikte potansiyel ilaçların keşfedilmesi için moleküler yapı özelliklerinin tahmini veya bir molekül uzayının içerisinde hedef özelliğe en uygun molekülün bulunması en önemli problemlerden biridir. Özellikle QSAR (Kantitatif Yapı-Aktivite İlişkisi) yaklaşımı ilaç keşfi ve yeni ilaç tasarımları için önemli araçlardan biridir. Bu alandaki öncü çalışmalara bakıldığında, Naive Bayes, K-En Yakın Komşu (KNN), Karar Ağaçları (Decision Trees), Destek Vektör Makinesi (SVM) gibi öne çıkan makine öğrenmesi yöntemleri, belirli veri kümelerinin yardımıyla moleküler yapı özelliklerinin tahmin edilmesi görevlerinde kullanılmaktadır [7]. Ek olarak, bir başka makine öğrenmesi yöntemi olan genetik programlama, ilaç tasarımı ve keşfi için önemlidir. [8]'de, bir ilacın organizmadaki emilimi, dağılımı, atılımı ve toksisite süreçlerini göz önünde bulundurarak, farmakokinetik parametrelerin tahmini görevinde genetik programlama kullanılmıştır. Genetik programlama çoğu makine öğrenmesi yöntemine göre daha iyi genelleme yeteneğine sahip olduğu için dikkat çekilmesi gereken bir yöntem olduğu belirtilmiştir.

Öte yandan, son gelişmelerin bir sonucu olarak, derin öğrenme yöntemleri, moleküler özellik tahmini ve molekül oluşturma görevlerinde ön plana çıkmaktadır. Literatürdeki araştırmalar incelendiğinde dört derin sinir ağı mimarisi diğer yöntemlere göre daha baskındır. Bu mimariler, özyinelemeli sinir ağları (recurrent neural networks), otomatik kodlayıcılar (autoencoders), üretici çelişkili ağlar (generative adversarial networks) ve takviyeli öğrenmedir (reinforcement learning) [9]. Bu tasarımlar genelde moleküllerin SMILES [10] dizi gösterimlerini kullanarak moleküler özelliklerin tahmininden üç boyutlu olarak görselleştirilmelerine, belirli protein alıcılarına bağlanmalarından çizge gramerlerinin çıkarılmasına kadar birçok görevde kullanılmaktadır. Bu görevlerde eğitilen modellerin yardımıyla, bilgisayar tabanlı yöntemleri kullanarak de novo ilaç tasarımı gibi sıfırdan istenen kimyasal ve farmakolojik özelliklere sahip yeni kimyasal bileşiklerin üretilmesinin otomatik bir şekilde yapılması amaçlanır [11] [12].

SMILES ile birlikte özyinelemeli sinir ağı mimarisine sahip modeller, molekül üretimi görevlerinde yaygın bir şekilde kullanılmaktadır. [13]'de, bu diziler ile özyinelemeli sinir ağı mimarisine sahip bir LSTM modeli, de novo ilaç tasarımı görevinde kullanılmıştır. Üretilen moleküllerin çoğu, sentetik erişilebilirlik skorlarına göre sentezlenebilir olduğu raporlanmıştır. Tasarlanan moleküllerin kimyasal geçerliliğini sağlamak, de novo ilaç tasarımıdaki en zorlayıcı kısımlardan biridir. Bu sorunu çözmek için literatürde Gramer

Varyasyonel Otomatik Kodlayıcılar (Grammar Variational AutoEncoders [14]) yaygın bir şekilde kullanılmaktadır. Oluşturulan moleküllerin kimyasal geçerliliğini artırmak için SMILES dizilerine gramer kısıtlamaları uygulanarak üretilen dizilerin geçerli olması amaçlanmıştır. Şüphesiz bu yaklaşımla güzel sonuçlar alınsa da ortaya çıkan moleküller her zaman geçerli değildir. Daha önceden de bahsedildiği üzere QSAR yaklaşımı ilaç üretimi ve keşfi için yaygın olarak uygulanmaktadır. [15]'te, ters QSAR yaklaşımı uygulanarak molekül üretme optimizasyon işlemleri uygulanmıştır. Burada, otomatik kodlayıcılar ile evrimsel sinir ağları ve özyinelemeli sinir ağları ayrı ayrı birleştirilerek de novo ilaç üretimi görevinde kullanılan bir model tasarlanmıştır ve belirlenen hedefler üzerinde istenilen molekülleri üretmek için bayes optimizasyonu [16] uygulanmıştır. Ayrıca yukarıda bahsedilen, SMILES dizilerini kullanarak geçerli moleküller üretilmesi için kullanılan varyasyonel otomatik kodlayıcılar bu çalışmanın da önemli bir bölümünü oluşturmaktadır. Varyasyonel otomatik kodlayıcı mimarilerine ek olarak, üretken çelişkili ağ mimarileri de son yıllarda [17] [18] popüler hale gelmiştir. [17]'da gerçekleştirilen deneylerde geçerli molekül üretme oranı diğer çalışmalara göre nispeten düşük olsa da ilgi çekici sonuçlar paylaşılmıştır. Buna karşın [18]'de, moleküllerin cezalı LogP skoru optimize edilip, çıktılarının %100'ünün geçerli bir molekül olduğu gösterilmiştir. Ayrıca üretken çelişkili ağ mimarilerine ek olarak, son zamanlarda çizge tabanlı yapay sinir ağlarının da gelişmesiyle, moleküllerin çizge yapılarını girdi olarak kullanan bu modeller etkileyici sonuçlar vermektedir. Çizge tabanlı bir üretici model olan GraphINVENT'de [19], kapılı çizge sinir ağı ile eğitilen modelin diğer üretici modellerden deneysel anlamda üstün olduğu paylaşılmıştır. Bir diğer araştırmada [20], SMILES dizileri yerine, bu dizilerin indirgenmiş çizge formunu kullanan modeller tasarlanmıştır. İndirgenmiş çizge, fonksiyonel grupların farmakoforik özelliklerini temsil eden süper atomlarla değiştirir. Bu yöntem, başta molekül üretimi olmak üzere birçok kimyasal bilişim görevlerine uygulanabilen alternatif bir derin öğrenme yaklaşımı sunmaktadır.

Bu tez çalışmasında da kullanılan takviyeli öğrenme, ilaç keşfi ve tasarımı alanında popülerliği artan bir diğer yaklaşımdır. Takviyeli öğrenmede, bu bağlamda, genellikle moleküllerin belirli yapısal bir özelliği ödül fonksiyonunda kullanılır ve bu fonksiyonun değerinin maksimize edilmesi amaçlanır. Takviyeli öğrenme özellikle derin öğrenme ile birlikte kullanıldığında birçok farklı tasarıma sahip olmuştur. [21]'de, en iyi politikayı bulmak için özyinelemeli sinir ağını A2C (Advantage Actor-Critic) ve PPO (Proximal Policy Optimization) gibi son teknoloji takviyeli öğrenme yöntemleriyle birleştirilerek, istenilen özellikler moleküllerin üretilmesi denenmiştir. Ancak çoğu çalışmada olduğu gibi burada da her üretilen molekül geçerli değildir. Molekül üretimi ve optimizasyonu için takviyeli öğrenme tasarımı kullanan bir başka yaklaşımda [22] ise SMILES için bağlamsız yeni bir gramer tasarlanmıştır ve bu takviyeli öğrenme sistemi ile

birleştirilmiştir. Bu şekilde garantili ve geçerli SMILES dizileri üretilecek şekilde tasarlanmıştır. Bundan farklı olarak [23]'de, yeni moleküller tasarlamak için bir dizi üreticinin üzerinde takviyeli öğrenme teknikleri kullanılmıştır. İstenilen özelliklerde moleküller çoğu zaman oluşturulsa da diğer çalışmalarda olduğu gibi her zaman geçerli molekül oluşturulmamıştır. Bu alandaki bir başka çalışmada ise [24], bu tezde de kullanılan MolDQN [3] yapısı, başka bir görev için değiştirilerek yeni bir model tasarlanmıştır. Buradaki görev, belirli protein alıcılarına kenetlenme skorlarını ödül fonksiyonunda kullanarak, başlangıç ligandlarını optimize etmektir. Deneysel olarak alınan sonuçlar, kıyaslanan temel yöntemlere göre deneysel olarak çok daha başarılı olduğunu göstermektedir. Yine bu tez çalışmasındaki yöntem benzer şekilde ancak diğer takviyeli öğrenme yöntemlerinden farklı olarak, çizge tabanlı yöntemler ile takviyeli öğrenme yaklaşımlarını birleştiren araştırmalar da mevcuttur [25], [26]. Bu çalışmalarda SMILES dizilerinden elde edilen özellikler yerine moleküllerin çizge yapısından yararlanılarak takviyeli öğrenme modelleri eğitilmiştir. Yapılan deneylerde, [25]'te %100 geçerlilik sağlanamazken, [26]'te üretilen tüm moleküllerin geçerli olduğu raporlanmıştır.

Bu tez çalışmasında yukarıdaki çalışmalara benzer şekilde, de novo ilaç tasarımı ve molekül optimizasyonu görevlerinde kullanılacak bir takviyeli öğrenme yaklaşımı uygulanmıştır. Temelde [3]'te tasarlanan takviyeli öğrenme (MolDQN) tasarımına sahip olduğu için üretilen kimyasal yapılar %100 geçerlidir. İlaç üretimi ve optimizasyonu görevlerinde yukarıdaki çoğu çalışmada kullanılan moleküller özelliklerin yanında bu çalışmada ayrıca sıfırdan belirli protein alıcılarına en iyi kenetlenen molekülün üretilmesi de amaçlanmıştır. Önde gelen ve deneysel anlamda diğer yöntemlere göre çok daha başarılı sonuçlar elde edilen MolDQN modeline yapılan eklemeler ile çok daha başarılı sonuçlar elde edilebilmiştir. Ayrıca, literatürdeki çalışmalardan farklı olarak, eğitim sürecini hızlandırmak için önceden eğitilmiş çizge tabanlı modeller kullanılmıştır. Ek olarak, bu modeller toksisite tahmini görevlerinde kullanıldığı için toksisite ile ilgili moleküller özelliklerin optimizasyonu görevlerinde önemli seviyede performans artışı sağlanmıştır.

### 3. ÖN BİLGİ

#### 3.1 SMILES

Basitleştirilmiş Moleküler Girdi Hattı Giriş Sistemi (SMILES, Simplified Molecular Input Line Entry System), kimyasal bilgi işleme için geliştirilmiş bir kimyasal gösterim sistemidir ve bu gösterimi ASCII karakterlerini kullanarak yapar. Bu sistem, moleküler çizge teorisi ilkelerine dayanır ve doğal kimyasal yapıya uygun bir dilbilgisi kullanarak kesin ve net bir yapı tanımlamasına izin verir [10]. Bu notasyon sistemi yüksek hızlı işlemlere uygun olduğu için günümüzde makine öğrenmesi başta olmak üzere pek çok yöntemde kullanılmaktadır.

SMILES, moleküler yapıyı, esasen kimyagerlerin bir molekülü tanımlamak için çizdiği iki boyutlu değerlik yönelimli resim olan bir çizge olarak belirtir ve üç boyutlu atom düzenini temsil etmez. SMILES notasyonu özünde, bir boşlukla biten bir dizi ASCII karakterdir. Hidrojen atomları gösterilmeyebilir (hidrojen bastırılmış çizgeler) veya dahil edilebilir (hidrojen tam çizgeler). Aromatik yapılar, Kekulê formuna tercih edilerek doğrudan belirtilir. Çoğu kimyasal yapı için SMILES üretme kuralları tanımlıdır. Bu kuralların ve gösterimlerin detaylarına ilgili kaynaktan [10] ulaşabilirsiniz.

#### 3.2 Morgan Parmak İzi Temsili

Molekülleri niteleyen ve belirli özelliklerini temsil eden birçok yöntem vardır. Bu yöntemlerden biri de literatürde yaygın olarak kullanılan Morgan/Dairesel (Circular) Parmak İzi'dir. Morgan algoritması temelde moleküllerin dairesel (belirli bir çapta) atomik komşular aracılığıyla temsil edilmesine dayanır. Bu tez çalışmasında, Morgan algoritmasının bir türevi olan Genişletilmiş Bağlantı Parmak İzi (Extended Connectivity Fingerprint, ECFP) kullanılmıştır [27]. ECFP'de kısaca, bağ düzenine ve atomik özelliklere göre moleküldeki her bir ağır atoma tanımlayıcı atanır, daha sonra bu tanımlayıcıları, belirli çaptaki komşu ağır atomlara atanan tanımlayıcılar ile birleştirerek molekülün temsili üretilir.

#### 3.3 Takviyeli Öğrenme

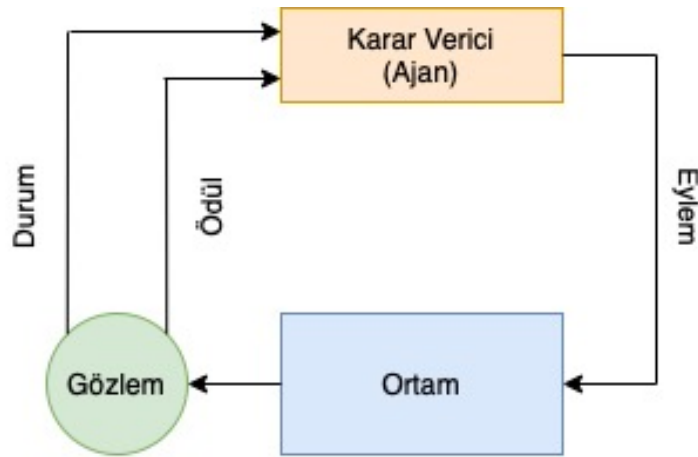
Takviyeli öğrenme, bir karar verici ajanın (agent) eylemlerinden ve deneyimlerinden geri bildirim alırken, deneme yanılma yoluyla etkileşimli bir ortamda optimal bir politika öğrenmesini sağlayan bir tür makine öğrenimi yaklaşımıdır. Başka bir deyişle, takviyeli öğrenme teknikleri, bu karar verici akıllı ajanların amaç fonksiyonunu (ödül fonksiyonu) en üst düzeye çıkarmak için bir ortamda nasıl hareket etmesi ya da nasıl

eylemler alması gerektiği ile ilgilenir [28]. Mevcut problemlere takviyeli öğrenme ile çözüm üretmek için en yaygın kullanılan yöntemlerden biri, ortamı Markov Karar Süreci olarak modellemektir. Bu şekilde, birçok takviyeli öğrenme yönteminde de olduğu gibi dinamik programlama tekniklerine benzeyen algoritmalar geliştirilmiştir. Takviyeli öğrenmede, çoğu ortam yapısından dolayı matematiksel olarak kesin bir şekilde modellenemediği için belirli yaklaşımlar ile bu problemler çözülmeye çalışılır [29].

Temel takviyeli öğrenme modeli aşağıdaki elementlerden oluşmaktadır.

- Durum (State): Karar vericinin ortamdaki durumlarının kümesidir.
- Eylem (Action): Karar vericinin bulunduğu duruma bağımlı veya bağımsız yapabileceği eylemlerin kümesidir.
- Ödül (Reward): Karar vericinin eylemini gerçekleştirdiğinde kazandıracığı veya kaybettireceği skaler değerdir.
- Politika (Policy): Karar vericinin içinde bulunduğu durumda hangi eylemlerin gerçekleştirilmesi gerektiğini belirler.

Daha önceden de bahsedildiği üzere buradaki temel amaç bulunan duruma göre en uygun (ödül değerini en optimal şekilde arttıracak) politikayı bulmaktır. Takviyeli öğrenme ortamında karar vericinin uyguladığı eylemler sonucu ortamdan gelecek skaler ödül değeri ve geçeceği yeni durum, belirli kurallar ile tasarlanır. Bu kurallar stokastik veya deterministik olabilir. Birçok çalışmada karar vericinin mevcut ortam durumunu gözlemleyebildiği kabul edilir. Buna tam gözlenebilirlik adı verilir. Ancak bunun karşıtı durumlar da söz konusudur ve kısmi gözlenebilirlik olarak adlandırılır. Takviyeli öğrenmenin nasıl işlediği genel hatlarıyla şekil 3.1’de gösterilmiştir.



Şekil 3.1: Takviyeli öğrenmenin genel işleyişi.



Denetimli öğrenme (supervised learning) ile takviyeli öğrenmenin en belirgin farkı, modeli eğitirken "etiketli" ve kesin bir geri bildirim ortamdan alınmıyor oluşudur. Ortamdan karar vericiye gönderilen bilgi kısmidir. Buna ek olarak, takviyeli öğrenmede zaman önemli bir faktördür. Özellikle sonlu sistemlerde bulunduğu duruma göre yapılan eylem tahminleri gelecekteki durumu ve potansiyel ödülü uzun vadede etkiler. İşin içine bu şekilde zaman faktörü girdiğinde Monte Carlo [30] veya dinamik programlama teknikleri nihai adımdan gelecek sonuca göre tüm zamanları değerlendirip en iyi eylemi alacak modeli tasarlamaya çalışır. Ancak, her ortam sonlu olmayabilir veya sonlanması için uzun zaman geçebilir ve bu da zaman açısından maliyetli olmasına neden olabilir. Bu zaman faktörünü göz önünde bulunduran bir yöntem olan Zamansal Fark (TD) Öğrenimi (Temporal Difference Learning) geliştirilmiştir. TD'de, Monte Carlo' dan farklı olarak her bir zaman adımındaki bilgiler ile bir adım sonraki sinyalin tahmini yapılarak eğitim işlemi gerçekleştirilir. Burada öğrenme işlemi gerçekleşirken geleceğe yönelik tahminler üzerinden yapıldığı için ortamdan alınan kısmi bir bilgi vardır ve bu nedenle aslında bir "yarı-denetimli" öğrenme yapılmaktadır. Buradaki denetim için kullanılacak veri değer fonksiyonun tahmini bir sonucu ile ödül verisinden geldiği için aslında "yarı-denetimli" olarak adlandırılmaktadır. Değer fonksiyonu, o andaki durumda bir eylemin ne kadar iyi olacağını tahmin eden bir fonksiyondur. Eğer bu değer fonksiyonları Monte Carlo yöntemindeki gibi tahmin edilmeksizin hesaplanırsa, yukarıda da bahsedildiği üzere nihai ödülün alınmasını beklemek gerekirdi. Bu nihai ödül alındıktan sonra, tekrar geri dönülmesi ve buna göre bu değer fonksiyonunun güncellenmesi gerekirdi. Bunun yerine TD, her bir adımda bu değer fonksiyonunu güncellenerek ilerler ve nihai ödülün alınmasını beklemez. TD'de gerçekleştirilen değer fonksiyonu güncelleme işlemi, denklem 3.1'de gösterilmiştir.

$$V(s) \leftarrow V(s) + \alpha(r + \gamma V(s') - V(s)) \quad (3.1)$$

Burada  $V(s)$ ,  $s$  durumundaki değer fonksiyonunu belirtir.  $\{ r + \gamma V(s') \}$ , TD hedef değerini,  $\alpha$  ise öğrenme oranını gösterir.  $s'$ , bir adım sonraki gelecek durumu belirtir. Dolayısıyla  $V(s')$ , bu durumundaki değer tahmini bir hesabıdır.  $\gamma$  ise 0 ile 1 arasında değer alan ve indirim faktörü (discount factor) olarak adlandırılan bir katsayıdır. Bu katsayı, TD öğrenimi uygulanırken, son adımlarda elde edilen değerlerin daha çok etkilemesini sağlar ve bu katsayı ne kadar 1'e yakın ise o kadar az indirim uygulanır.

### 3.4 Q Öğrenme

Literatürde en yaygın olarak kullanılan takviyeli öğrenme algoritmalarının başında Q öğrenme gelmektedir. Ortam hakkında hiçbir bilgi veya gözlemlenebilirlik olmadan çevrimiçi bir şekilde öğrenmesini sağlayan bir yöntemdir. Herhangi bir ortam modeli

gerektirmedeği için modelsiz (model-free)'dir. Takviyeli öğrenme kısmında (bölüm 3.3) bahsedildiği üzere ana amaç karar verici ajanların buldukları duruma göre en iyi kararları almasını sağlamaktır. Bir başka deyişle en iyi politikayı bulmak amaçlanır. Q öğrenme yönteminde, anlık olarak toplanan ödül değerlerini, yani eğitim verilerini kullanarak bir değerlendirme fonksiyonu (Q function) eğitilir. Daha sonra bu fonksiyon yardımıyla optimal politika belirlenir. Bu algoritmanın ana hedefi, bir sonraki eylemi incelemek, uyguladığı eylem için alacağı ödülü belirlemek ve bu ödülü en üst düzeye çıkaracak şekilde hareket etmektir.

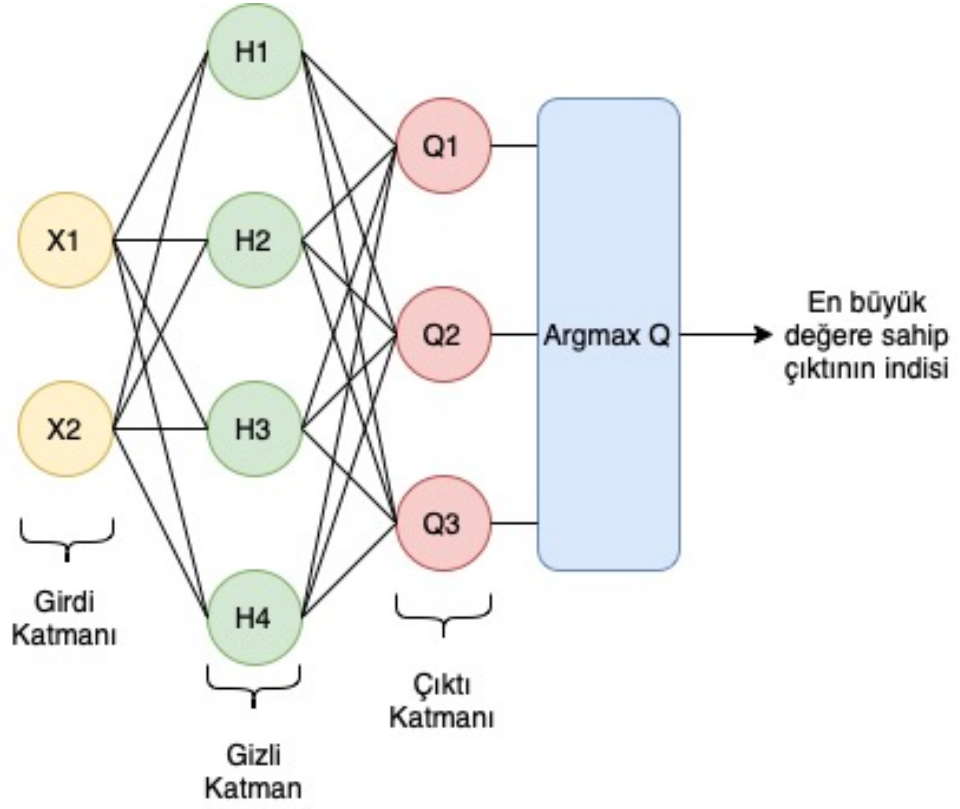
Q öğrenme algoritması, Q çizelgesi olarak adlandırılan durum ve ödül değerlerini sözlük yapısında saklayan bir sistem üzerine kurulmuştur. Belirli bir tekrar süresince (bölüm sayısı olarak adlandırılır) eylemlerini gerçekleştirip hareket eden karar verici ajan, bu hareketler sırasında buradaki sözlük yapısına sahip çizelgede durum değerlerini, gerçekleştirilen eylemler doğrultusunda en iyi ödül değerleri ile eşleştirir. Belirlenen tekrar (bölüm) sayısı tamamlandığında, nihai durumdaki çizelge optimal sonuçları gösterir. Bir başka deyişle buradaki Q çizelgesi politikayı oluşturur. Hangi durumda hangi eylemin seçileceği bu çizelgeye bakılarak karar verilebilir. Takviyeli öğrenmede ortam Markov Karar Süreci olarak modellendiği için bu yapı Markov özelliğine sahiptir. Bu özelliğe göre, geçmiş durumlar veya eylemler arasında herhangi bir bağımlılık yoktur, ortamın sadece bir adım için etkinliği tanımlanır [31].

### 3.4.1 Derin Q öğrenme

Q öğrenme bölümünde de bahsedildiği üzere, Q öğrenme, Q çizelgesi olarak adlandırılan durum ve bu durumda alınan eylemlerin sonucunda alınan ödül değerlerinin tutulduğu bir yapının üzerine kurulmuştur. Ancak bu yapı birçok problemde yetersiz kalabilir. Örneğin en basitinden, sonsuz veya çok büyük bir durum uzayının olduğu durumlarda bu yapıyı saklayabileceğimiz bir bellek yoktur.

Bu sorunu çözmek için Q çizelgesi yerine bir derin yapay sinir ağı modeli kullanan yöntemler, derin Q öğrenme olarak adlandırılır. Belirli andaki durum bilgileri bu modele girdi olarak verilip alınan çıktılar eylemleri seçmek için kullanılmaktadır. Genelde bu modellerin çıktıları, eylem kümesi kadardır ve çıktı düğümlerinin tahminine göre en büyük değere sahip olan düğümün indisi hangi eylemin yapılacağını belirler. Şekil 3.2'de bu sürecin bir örneği görselleştirilmiştir.

Burada ortaya çıkan bir problem ise bu yapay sinir ağlarının nasıl eğitileceğidir. TD, yöntemi kullanılarak eğitilen bu modeller sürekli her adımda güncellenirse yapısı gereği stabil bir öğrenme olmayacaktır. Bunu çözmek için birçok yöntem geliştirilmiştir. Bu yöntemlerden en yaygın olanı bir tekrar arabellek (replay buffer) tanımlayarak belirli aralıklarla bu modeli buradaki veriyi kullanarak eğitmektir. Tekrar arabelleğine, eğitim aşamasında keşfedilen durum, eylem, ödül ve gelecek durum bilgileri kaydedilir. Daha



Şekil 3.2: Derin Q öğrenmede kullanılan bir derin yapay sinir ağının örneği.

sonra, belirlenen aralıklarla bu arabellekten rastgele bir şekilde alınan ve belirli bir büyüklüğe sahip veriler ile bu model eğitilir. Eğitim aşamasında, belirlenen nihai bölüm tamamlandıktan sonra burada öğrenilen model, takviyeli öğrenmede amaçlanan politikayı oluşturur [32].

## 4. YÖNTEMLER

Bu tez çalışmasında kullanılan yöntem temel olarak iki yapının birleştirilmesinden oluşmaktadır. [3] çalışmasından elde edilen Derin Takviyeli Öğrenme mimarisine sahip MolDQN modeli ilk yapıyı oluştururken, bu tasarıma eklediğimiz çizge sinir ağı mimarisine sahip modeller ise ikinci yapıyı oluşturmaktadır. Aşağıdaki alt başlıklarda ilk olarak, kullanılan MolDQN yapısının detaylarından bahsedilmiştir, ardından çizge tabanlı modellerin yapısı, matematiksel ve teknik detayları ile birlikte açıklanmıştır. Son olarak bu iki yapının nasıl birleştirilip tek bir model ortaya çıkarıldığından bahsedilmiştir.

### 4.1 Derin Takviyeli Öğrenme Modeli (MolDQN)

Takviyeli öğrenme, kümülatif ödülü en üst düzeye çıkarmak için karar verici ajanların bir ortamda nasıl harekete geçmeleri gerektiği ile ilgili bir makine öğrenimi alanıdır. Derin Takviyeli Öğrenme ise Takviyeli Öğrenme ile derin öğrenmeyi birleştiren bir makine öğrenimi alt alanıdır. Derin Takviyeli Öğrenme algoritmaları (Takviyeli Öğrenme'nin aksine) çok büyük girdilerin gerektiği ortamlarda veya sonlu olmayan durumlara sahip ortamlarda bir hedefi optimize etmek için hangi eylemlerin gerçekleştirileceğine karar vermede daha kullanışlı olabilmektedir. Derin Takviyeli Öğrenme, bilgisayarla görü, doğal dil işleme, eğitim, ulaşım, finans, robotik, video oyunları, sağlık vb. çeşitli alanlardaki uygulamalarda kullanılmaktadır [33]. Bu çalışma da sağlık alanındaki (ilaç optimizasyonu ve üretimi) bir araştırmaya örnek olarak gösterilebilir.

MolDQN yapısında, molekül optimizasyonu süreci bir Markov Karar Süreci (MDP) olarak formüle edilmiştir ve bir molekülün değiştirilmesi veya optimize edilmesi bu mimaride adım adım yapılan bir süreçtir. Her adım üç kategoriden birine girer: atom ekleme, bağ ekleme ve bağ çıkarma. Matematiksel olarak, MDP aşağıdaki gibi ifade edilebilir:

$$MDP(S, A, \{P_{sa}\}, R) \quad (4.1)$$

Burada  $S$ , durum uzayını belirtmektedir. Her bir durum, iki elemanlı demetlerden oluşmaktadır. Bu demetin bir elemanı geçerli bir molekül, diğer elemanı ise o ana kadar uygulanan adım sayısını belirtir. Başlangıç durumu için molekül, boş eleman veya belirli bir başlangıç molekülü olabilir. Başlangıç durumunda o ana kadar uygulanan adım sayısı ise sıfırdır.  $A$ , her eylemin belirli bir molekül için geçerli bir modifikasyon olduğu eylem uzayını belirtir. Her modifikasyon, daha önce de bahsedildiği gibi üç kategoriden (atom ekleme, bağ ekleme ve bağ çıkarma) birine aittir.  $\{P_{sa}\}$  durum

geçiş fonksiyonunu belirtir ve deterministik olarak tanımlanmıştır. Örneğin, bir atom ekleme eylemi gerçekleşeceği zaman (eğer geçerli bir işlemse) %100 olasılığında gerçekleşir. Son olarak  $R$ , ödül fonksiyonunu ifade etmektedir. Bu tasarımda, ödül genellikle molekülün bir özelliğidir ve her adımda bir indirim faktörü kullanılarak bir ödül verilir. Her bölüm sonunda da bu ödül değerlerinin indirim faktörleri ile birlikte kümülatif değeri hesaplanır. Böylece, bölüm sonuna doğru alınan ödül değerleri doğal olarak toplam ödülü daha çok etkilemiş olur.

Bu tasarımdaki ortam, yukarıda tanımlanan molekül modifikasyonlarıdır. Oluşan her bir yeni molekül o anki durumu belirtir ve amacımız, her durum için en iyi eylemi seçerek gelecekteki ödülleri en üst düzeye çıkarabilecek bir  $\pi$  politikası bulmaktır. Politika  $\pi$  aşağıdaki formüldeki [4.2] gibi tanımlanır.

$$\pi^*(s) = \operatorname{argmax}_a Q^{\pi^*}(s, a) \quad (4.2)$$

Kısaca,  $s$  durumunda  $a$  eyleminin gelecekteki ödülleri tahmin eden  $Q(s, a)$  fonksiyonu maksimize edilmelidir.  $Q(s, a)$  fonksiyonu aşağıdaki [4.3] gibi tanımlanmıştır.

$$Q^{\pi}(s, a) = \mathbb{E}_{\pi} \left[ \sum_{n=t}^T r_n \right] \quad (4.3)$$

Burada  $r_n$ ,  $n$  adımıdaki ödülü belirtir ve  $\mathbb{E}_{\pi}$ ,  $\pi$  ile ilgili bir beklenti almayı belirtir. Kimyasal uzayın boyutu nedeniyle, burada tasarlanan MDP’de uygulanan adımların sayısı sınırlıdır. Yani eğitim aşamasında belirli sayıda eylem yapılmaktadır. Bu nedenle adım boyutunu  $T$  belirleyerek bir molekülün değişimi kontrol edilmiştir. Derin  $Q$ -öğrenme olarak adlandırılan bu teknikte,  $Q$  fonksiyonunu temsil etmek için derin yapay sinir ağı kullanılır. Sinir ağı fonksiyonu tahmincisi, parametreleştirilmiş  $Q$ -değer fonksiyonu ( $Q(s, a; \theta)$ ) olarak tanımlanır, burada  $\theta$  optimize edilecek parametredir. Bu tahminciye eğitmek için kullanılan kayıp fonksiyonu aşağıdaki denklemde 4.4 tanımlanmıştır.

$$l(\theta) = \mathbb{E}[f_l(y_t - Q(s_t, a_t; \theta))] \quad (4.4)$$

Burada  $y_t$ ,  $r_t + \max_a Q(s_{t+1}, a; \theta)$  ve  $f_l$  ise Huber hata fonksiyonunu belirtir [34].

Yukarıda da bahsedilen  $Q$  fonksiyonu tahmincisi bu tasarımda bir derin yapay sinir ağı mimarisinden oluşmaktadır. Farklı molekül temsilleri ve farklı görevlerde birçok farklı model eğitilmesine rağmen temel model tasarımı hepsinde aynıdır. Bu derin yapay sinir ağı modeli, ileri beslemeli tam bağlantılı bir yapay sinir ağıdır ve 4 gizli katmandan oluşur. Farklı molekül temsilleri için sadece giriş katmanı değişmektedir, geri kalan katmanlar her model için aynı sayıda ve şekilde kullanılmıştır. Gizli katmanların boyutları sırasıyla 1024, 512, 128, 32’dir ve tüm bu katmanlar Rectified Linear Unit

(ReLU) aktivasyon fonksiyonuna sahiptir. Giriş katmanı, bahsedildiği gibi molekül temsillerinin boyutuna göre değişmektedir.

Bir takviyeli öğrenme karar vericisi, kümülatif olarak en fazla ödülü toplayabilmek için geçmişte denediği ve ödül üretmede etkili bulunduğu eylemleri seçmek zorundadır. Bunu yaparken eğer hiç keşif yapmadan, sadece geçmiş deneyimlerinden faydalanırsa (sömürü), açgözlü bir yaklaşım sergiler ve bu da doğal olarak optimal bir yaklaşım değildir. Bu yüzden, daha önce hiç denemediği eylemleri deneyerek (keşif) bulunduğu ortamda daha iyi sonuçlar alabilmesi için önemlidir. Buradaki ikilem, yalnızca ne keşfin ne de sömürünün görevinde başarısızlığa uğramadan yürütülemeyeceğidir [35]. Bu sorunu çözmek için takviyeli öğrenme modellerinde yaygın olarak kullanılan bir keşif stratejisi olan epsilon-açgözlü ( $\epsilon$ -açgözlü) yöntemi geliştirilmiştir. Geliştirilen bu yöntem MolDQN tasarımında da kullanılmaktadır.  $\epsilon$ -açgözlü, eylemleri belirli bir oranda rastgele seçerek keşif ve sömürüyü dengelemeye çalışan basit bir yöntemdir.  $\epsilon$  gibi küçük bir olasılıkla, o andaki durumda mevcut eylemler arasından rastgele bir eylem seçilerek ortamın keşfi sağlanır. Sömürü için ise  $1 - \epsilon$  olasılığıyla geçmiş deneyimleri kullanarak eğittiğimiz modelin tahmin ettiği en iyi eylem seçilir.

Öte yandan,  $\epsilon$ -açgözlü, katlanarak daha fazla veri gereksinimlerine yol açabileceğinden ve değer fonksiyonu tahmininin belirsizlik seviyesini göz önünde bulundurmadan, genellikle daha değersiz olduğu bilinen durumlarda keşif çabasını boşa harcayabilir ve ortamı derinlemesine bir şekilde keşfedemeyebilir. Bu sorunu çözmek için [3]'te, [36]'te geliştirilen önyüklemeli-DQN (bootstrapped-DQN) yöntemi kullanılmıştır. Bu yöntem, rasgele değer fonksiyonlarını kullanarak derinlemesine bir keşif yapabilir. Önyüklemeli-DQN konseptinin biraz daha detayına inerek; takviyeli öğrenme modeli  $H$  adet bağımsız  $Q$  fonksiyonlarından oluşur ve her bir  $Q$  fonksiyonu için ayrı bir çıktı katmanı (OSBAND çalışmasında kafa olarak adlandırılan) vardır. Bu haliyle aslında çok görevli yapay sinir ağına benzemektedir ve bu nedenle çıktı katmanının boyutu  $H$  değeri kadardır. MolDQN çalışmasında bu değer, yapılan parametre optimizasyonu sonucunda 12 olarak ayarlanmıştır (bu tez çalışmasında da bu değer aynen kullanılmıştır). Bu konseptteki her bir  $Q$  fonksiyonu, ortamdaki elde edilen verilerin farklı bir alt kümesini kullanarak eğitildiği için daha derin bir keşif yapabilmektedir. Eğitim aşamasında her bölüm için eşit olasılıkla rastgele bir  $Q$  fonksiyonu (kafa) seçilerek karar verme işlemi gerçekleştirilir. Bu modelde  $\epsilon$ -açgözlü ile önyüklemeli-DQN yöntemi birleştirilerek kullanılmaktadır. Eğitim aşamasında  $\epsilon$  değeri 1.0 olarak başlatılıp, 0.01 değerine doğrusal olarak azalacak biçimde uygulanmıştır.

## 4.2 Çizge Sinir Ağı Modelleri

Bu çalışmada, Çizge Evrişimsel Ağı (GCN) ve Çizge Dikkat Ağı (GAT) olmak üzere iki farklı çizge sinir ağı tasarımı kullanılmıştır. Bu iki tasarım, bu alanda yaygın olarak

kullanıldığından ve deneysel sonuçları genellikle diğer çizge sinir ağı yapılarına göre daha iyi olduğu için bu modellerin kullanılması uygun görülmüştür [37]. Ayrıca ek olarak, önceden eğitilmiş bu iki mimariye (GCN, GAT) sahip çizge tabanlı modeller kullanarak eğitim sürecinin süresini azaltmak da hedeflerimizden biridir. Çalışmadaki bir diğer motivasyon ise toksisite ile LogP (optimize edilmesi amaçlanan molekül özelliklerinden biri) arasındaki ilişkinin araştırılmasıdır. Bu nedenle, Tox21 Challenge veri kümesini [5] kullanarak moleküler toksisite tahmini için geliştirilen önceden eğitilmiş çizge sinir ağı mimarisine sahip modeller tercih edilmiştir. Böylece, hem önceden eğitilen modellerin çıktılarını önerilen modele entegre ederek eğitim aşamasının süreci hızlandırılmıştır, hem de LogP ile toksisite arasındaki korelasyonu kullanarak daha iyi sonuçlar alınması hedeflenmiştir.

#### 4.2.1 Çizge evrişimsel ağı

Çizge Evrişimsel Ağı ya da kısaca GCN, doğrudan çizgeler üzerinde çalışabilen ve yapısal bilgilerinden yararlanabilen bir tür evrişimsel sinir ağıdır. GCN'nin arkasındaki temel ilke, her düğüm için, düğümün kendi özelliğinin yanı sıra tüm komşularından özellik bilgisinin elde edilebilmesidir. GCN'nin temel yapısı, aşağıdaki (GCN) katman bazında yayılma (layer-wise propagation) kuralına dayalıdır [38].

$$H^{(l+1)} = \sigma(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} H^{(l)} W^{(l)}) \quad (4.5)$$

$H^{(l)}$  ağın  $l^{th}$  katmanını belirtmektedir.  $\sigma$ , sigmoid aktivasyon fonksiyonunu belirtir.  $W$  bu katman için ağırlık matrisini belirtmektedir. Komşu matrisler (adjacency matrices)  $A$  ile gösterilir. Çizge evrişimlerinde eski temsili korumak için her bir düğüm ve kendisi arasında fazladan kenarların (edge) eklendiği varyant,  $\tilde{\cdot}$  ile gösterilmiştir.

İlk katmandaki (giriş katmanı) girdi,  $H$  ( $H^{(0)}$ )  $N \times D$  şekline sahiptir. Burada  $N$  düğüm sayısını ve  $D$  girdi özniteliklerinin sayısını belirtir. Ayrıca,  $N \times F$  şekline sahip, düğüm düzeyinde bir temsil çıktısı üretmek için birden çok katman birbirine zincirlenebilir. Burada  $F$  ise düğüm özniteliklerinin çıktı sayısını gösterir.

#### 4.2.2 Çizge dikkat ağı

Yerel çizge yapısı ile düğüm seviyesindeki özniteliklerin birleştirilmesi, düğüm türünü tahmin etme problemlerinde yaygın bir şekilde kullanılmaktadır [38]. Bununla birlikte GCN, öğrenme aşamasında düğüm öznitelikleri ile çizge yapısına bağlı (structure-dependent) bir şekilde ağırlık vektörlerini topladığı için genelleme yeteneği sınırlanabilir. Böylece, aşırı öğrenme (overfitting) sorunu ortaya çıkabilir. Öte yandan, Çizge Dikkat Ağı ya da kısaca GAT, GCN'den farklı olarak özgün bir toplama yöntemi sunar [39]. Bu mimarideki dikkat mekanizması, GAT'ın komşu özniteliklerini ağırlandırarak öznitelik

bağımlı ve yapıdan bağımsız normalleştirme uygulayarak çalışmaktadır. Bir başka deyişle, GAT ve GCN arasındaki temel fark, düğümün tek atlamalı komşusundan (one-hop neighborhood) gelen bilgilerin toplanma biçimidir. GCN'deki bir çizge evrişim işlemi, komşuların düğüm özniteliklerinin normalleştirilmiş toplamını vermektedir 4.6.

$$h_i^{(l+1)} = \sigma \left( \sum_{j \in N(i)} \frac{1}{c_{ij}} W^{(l)} h_j^{(l)} \right) \quad (4.6)$$

$N(i)$  düğümün tek atlamalı komşularının kümesini belirtmektedir.  $c_{ij}$ , çizge yapısına dayalı bir normalleştirme sabitidir ve aşağıdaki formülde (4.7) nasıl hesaplandığı gösterilmiştir.

$$c_{ij} = \sqrt{|N(i)|} \sqrt{|N(j)|} \quad (4.7)$$

$\sigma$  sigmoid aktivasyon fonksiyonunu belirtir ve  $W^{(l)}$  düğüm bazında öznitelik dönüşümü için paylaşılan bir ağırlık matrisidir.

GAT tarafından kullanılan dikkat mekanizması, evrişimli ağın aksine normalleştirme işlemi, yukarıda da gösterildiği üzere statik olarak gerçekleştirmez. Düğüm temsilini (node embedding) hesaplamak için uygulanan işlemler aşağıda verilmiştir. Bu işlemler,  $l$ ' inci katmanın düğüm temsillerini ( $h_i^{(l)}$ ) kullanarak  $l + 1$ ' inci katmanın düğüm temsilinin ( $h_i^{(l+1)}$ ) nasıl hesaplandığını gösterir.

$$z_i^{(l)} = W^{(l)} h_i^{(l)} \quad (4.8)$$

Burada  $W^{(l)}$  öğrenilebilir ağırlık matrisini belirtmektedir. (4.8) denklemi,  $h_i^{(l)}$  ve  $W^{(l)}$ 'yi içeren alt katman temsilinin doğrusal bir dönüşümünün nasıl hesaplandığını gösterir.

$$e_{ij}^{(l)} = \text{LeakyReLU}(\vec{a}^{(l)T} (z_i^{(l)} || z_j^{(l)})) \quad (4.9)$$

(4.9) denkleminde,  $||$  birleştirme işlemi belirtirken,  $\vec{a}^{(l)}$  öğrenilebilir ağırlık vektörünü belirtir. Bu denklemde, iki komşu düğüm arasında ikili bir dikkat puanı hesaplanır. Önce  $z$  temsilleri birleştirilir, ardından bu birleştirmenin sonuçları ile  $\vec{a}^{(l)}$  arasında bir skaler çarpım alınır. Son olarak, LeakyReLU aktivasyon fonksiyonu uygulanır. Bu işlem, ek dikkat (additive attention) olarak da adlandırılır.

$$a_{ij}^{(l)} = \frac{\exp(e_{ij}^{(l)})}{\sum_{k \in N(i)} \exp(e_{ik}^{(l)})} \quad (4.10)$$

(4.10) denkleminde ise her bir düğümün gelen kenarlarındaki dikkat puanları softmax aktivasyon fonksiyonu kullanılarak normalleştirilir.



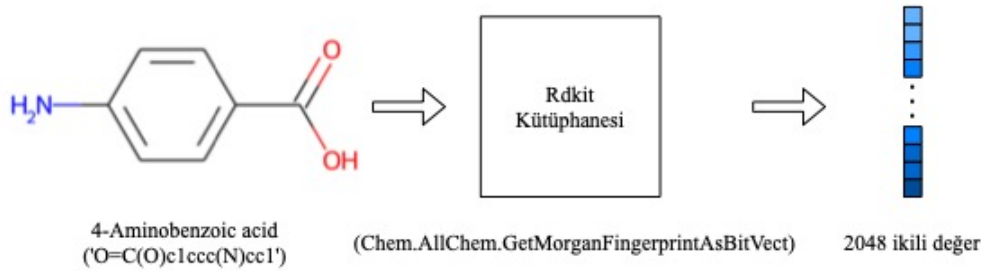
$$h_i^{(l+1)} = \sigma \left( \sum_{j \in N(i)} a_{ij}^{(l)} z_i^{(l)} \right) \quad (4.11)$$

GCN tasarımına benzer olarak, (4.11) denkleminde, komşu düğümlerden gelen temsiller toplanır ve dikkat puanlarına göre ölçeklenir.

Bu şekilde statik bir toplama işlemi gerçekleşmez ve model daha genelleyici bir biçimde, çizge verisinden çıkarım yaparak öğrenebilir.

### 4.3 Önerilen Model

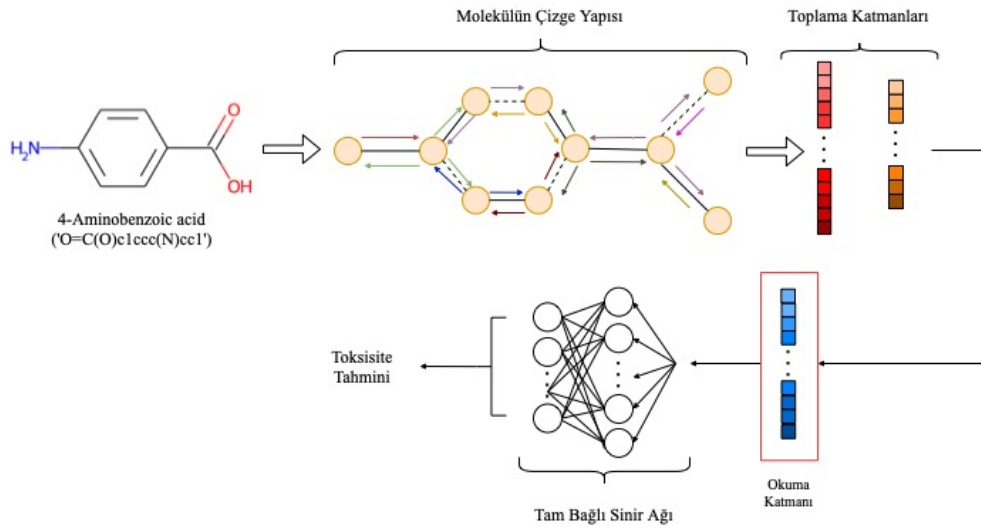
Takviyeli öğrenmede model, algoritmanın yapısı gereği bölümler halinde eğitilmektedir. Her bölüm de kendi içinde adımlardan oluşmaktadır. Daha önceden de bahsedildiği üzere eğitim süreci adım adım bir şekilde ortamın kuralları dahilinde yeni moleküller oluşmaktadır. Oluşan yeni moleküller ile son adımda (bölüm biterken) en yüksek ödül değerine ulaşılması hedeflenmektedir. Algoritmanın iyi bir şekilde çalışabilmesi için her adımda ortaya çıkan molekülün (anlık durum bilgisi) en iyi şekilde betimleyen temsilinin oluşturulması gereklidir. Temel yöntemde bu işlem Morgan Parmak İzi kullanılarak yapılmaktadır. Ortamdan gözlemlenen durum (molekül) SMILES molekül temsili olarak gelir. Rdkit kütüphanesi [40] kullanılarak bu SMILES temsili bir molekül nesnesine çevrilir. Daha sonra bu molekül nesnesi, yine rdkit kütüphanesinden yararlanılarak (GetMorganFingerprintAsBitVect metodu ile) 2048 uzunluğunda ve ikili değere sahip bir vektör elde edilir. Bu vektör, o andaki molekülün temsili olarak derin Q öğrenme yönteminde kullanılmak üzere bir ileri beslemeli yapay sinir ağına girdi olarak verilir. Bu ara işlem bir örnekle şekil 4.1’de gösterilmiştir.



Şekil 4.1: Morgan Parmak İzi'nin elde edilme süreci.

Bundan farklı olarak önerilen yöntemde, ilk olarak gözlemlenen molekül (durum) Derin Çizge Kütüphanesi (Deep Graph Library, DGL) kullanılarak çizge yapısına dönüştürülür. Elde edilen SMILES temsili, dgllife.utils.smiles\_to\_bigraph metodu ile çizge yapısına dönüştürülmektedir. Daha sonra bu çizge yapısı, çizge tabanlı modellere girdi olarak verilir. Çizge modelleri DGL kütüphanesi kullanılarak oluşturulduğu için direkt olarak

çizge yapıları modellerin girdi katmanında kullanılabilir. Daha önceden de bahsedildiği üzere, bu tasarımda eğitim aşamasını hızlandırmak için önceden eğitilmiş modeller kullanılmıştır. Bu modeller toksisite tahmini yapmak için eğitilmiştir. Bu nedenle doğal olarak, modelin çıktıları molekül temsili vermez. Ancak, bu modeller toksisite tahmini için eğitilirken moleküllerin temsillerini de çizge yapılarını kullanarak öğrenmektedir. Bu temsilleri elde etmek için toplama katmanlarından sonra gelen okuma katmanında üretilen vektörü kullanabiliriz. Nitekim buradaki çalışmada da bu yapılmıştır. Bu işlem şekil 4.2’de bir örnekle gösterilmiştir. Kırmızı dikdörtgen ile gösterilen okuma katmanı, çizge tabanlı modellerden alınan molekül temsillerini belirtir.



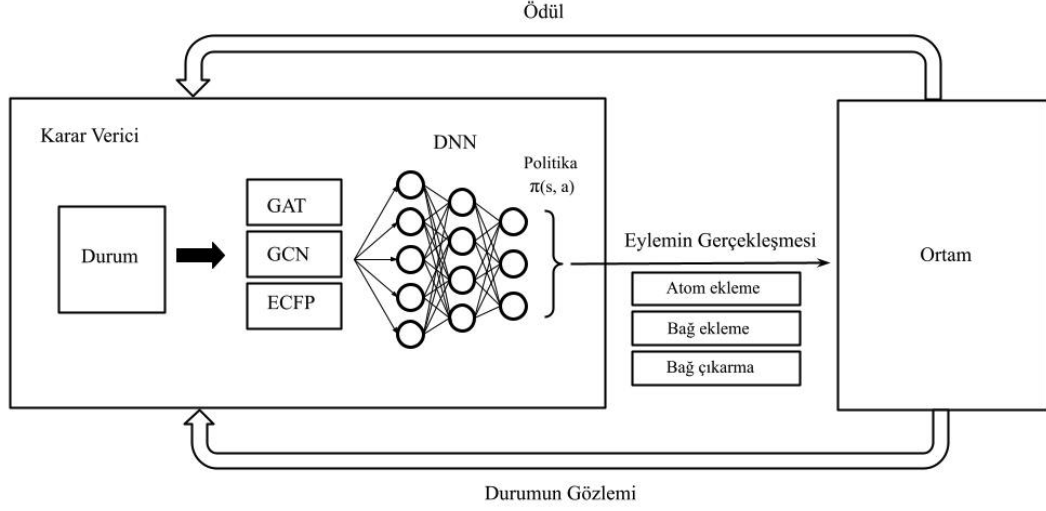
Şekil 4.2: Çizge modellerinden elde edilen molekül temsilleri.

Yukarıda bahsedilen bu molekül temsilleri derin Q öğrenme yönteminde, tam bağlı ileri beslemeli yapay sinir ağında girdi olarak kullanılmak üzere elde edilmiştir. Önerilen modelin genel süreci şekil 4.3’te görselleştirilmiştir.

Bu şekilde genel olarak tasarlanan modelin nasıl çalıştığı özetlenmiştir. En iyi politikayı bulmak için eğitilen bir tam bağlı ileri beslemeli yapay sinir ağından, o andaki duruma göre en uygun eylem çıktıları, tasarlanan ortamda uygulanmaktadır. Bulduğu duruma göre uygulanacak eylemler, daha önceden de bahsedildiği üzere atom ekleme, bağ ekleme ve bağ çıkarma işlemleridir ve bu işlemler belirlenen kimyasallar kurallara göre uygulanır. Bu sayede nihai adımda her zaman geçerli bir molekül üretilir. Ortamdan gözlemlenen durumlar her adımda üretilen moleküllerdir ve bu moleküller kullanılarak nihai adımda en iyi molekülün üretilmesi amaçlanmıştır. Gözlenen duruma ek olarak, kalan adımların sayısı da durum bilgisini bize vermektedir.

GAT, GCN ve ECFP olarak gösterilen modüller ortamdan gözlenen moleküllerin temsillerini üretmek için kullanılmaktadır. GAT ve GCN modellerinden nasıl molekül

temsillerinin alındığı önceki bölümlerde detayları ile aktarılmıştır. Temel yöntem olarak ise ECFP ismiyle gösterilen Morgan Parmak İzi temsilleri ile çizge tabanlı modellerin okuma katmanlarından alınan molekül temsillerini kullanarak eğittiğimiz modeller yapılan deneyler ile kıyaslanmıştır. Bu deneylerin detaylarına bölüm 5’te yer verilmiştir.



Şekil 4.3: Önerilen modelin işleyişi.

## 5. DENEYSEL SONUÇLAR

Bu bölümde, ilaç optimizasyonu ve üretimi için tasarlanmış modelleri kullanarak yapılan deneylerin sonuçları sunulmaktadır. Önerilen model ile kıyaslamak için temel yöntem olarak [3]'te yapılan (Morgan Parmak İzi ile) üç deney seçilmiştir. [3]'te yapılan deneylerin sonuçları literatürdeki diğer yöntemlere kıyasla çok daha başarılı olduğu için önerilen model ile kıyaslarken temel yöntem olarak kullanılmıştır. Bu deneylere ek olarak [41]'da yapılan ve hedef protein alıcılarına ne kadar iyi kenetlendiğini ölçen bir başka deney de yapılmıştır. Yine aynı şekilde temel yöntem olarak Morgan Parmak İzi gösterimini kullanan model tercih edilmiştir. Son deneyde ise hedef olarak belirlenen protein alıcılarına en iyi şekilde kenetlenen moleküllerin (ilaçların) üretilmesi için çalışılan deneyde öğrenilen Q fonksiyonu kullanılarak, rastgele başlangıç moleküllerinin optimize edilmesi hedeflenmiştir. Burada kıyaslanabilecek herhangi bir temel yöntem olmadığı için rastgele seçilen eylemler ile optimize edilen moleküller arasında bir kıyas yapılmıştır.

Uygulamanın ve karşılaştırmanın daha kolay olduğu, [3]'te gerçekleştirilen üç deney sırasıyla, QED (İlaç Benzerliğinin Kantitatif Tahmini) değerinin optimizasyonu (tek amaçlı), benzerlik kısıtlaması ile cezalı LogP değerinin optimizasyonu ve benzerlik kısıtlaması ile QED değerinin optimizasyonu (çok amaçlı) şeklindedir. Kenetlenme deneylerinde ise literatürde yaygın olarak kullanıldıkları ve optimizasyon işlemi için farklı zorluk seviyelerine sahip olduklarından dolayı üç farklı hedef protein alıcısı tercih edilmiştir (F2, PPAR, JAK2). Ayrıca, kenetlenme deneylerinin sonucunda elde edilen en iyi modelin (GAT) öğrenilen Q fonksiyonunu kullanarak, farklı ve rastgele seçilen başlangıç noktalarının optimize edilmesi sonucu üretilen moleküller tez çalışmasındaki son deneyi oluşturmaktadır. Bu deneylerin sonuçları ve hangi parametre değerlerinin kullanıldığı alt bölümlerde ayrıntılı olarak ele alınmıştır.

İlaçların çizge tabanlı temsillerini oluşturmak için Derin Çizge Kütüphanesi'nden (DGL) yararlanılarak önceden eğitilmiş GAT ve GCN modelleri kullanılmıştır [42]. Daha önce bahsedildiği gibi, her iki model de Tox21 Challenge veri kümesindeki toksisiteyi tahmin etmek için eğitilmiştir ve bu modellerin okuma katmanlarındaki ağırlık vektörleri, ilaçların temsili olarak kullanılmıştır. Bu önceden eğitilmiş modellerde kullanılan katmanların boyutları önceden belirlendiği için değerleri değişmemektedir. Temsil vektörünün boyutu GAT modeli için 64 iken GCN modeli için 128'dir. Bilindiği üzere, bu temsiller ile eğitilen modeller temel yöntem olarak belirlenen Morgan Parmak İzi temsiller ya da kısaca ECFP (Genişletilmiş Bağlantı Parmak İzi) [27] ile eğitilen modeller kıyaslanmıştır. Bu ECFP vektörü ise 2048 ikili yapısal öznitelik içermektedir.

Tüm deneyler, aynı parametre ayarları (bölüm 5.1) kullanılarak eğitilen modeller ile yapılmıştır.

## 5.1 Model Ayarları

### 5.1.1 Derin Q ağı modeli (DQN)

Bölüm 4.1’de açıklanan MolDQN mimarisinde kullanılan bu tam bağlı ileri beslemeli derin yapay sinir ağı modelinin amacı, girdi olarak verilen durum bilgisine göre en iyi eylemi tahmin etmektir. Daha önceden de bahsedildiği üzere bu model bir girdi katmanı, dört gizli katman ve son olarak bir tane çıktı katmanından oluşmaktadır. Girdi katmanı kullanılan molekül temsiline göre değişmektedir. ECFP için 2048, GCN için 128 ve GAT için 64 olarak ayarlanmıştır. Gizli katmanlarındaki düğüm sayısı ise sırasıyla 1024, 512, 128 ve 32 olarak ayarlanmıştır. Ayrıca, aşırı uymanın (overfitting) önüne geçmek için geliştirilen bir yöntem olan yığın normalleştirme (batch normalization [43]) katmanları bu modeli eğitirken her gizli katman sonrası uygulanmıştır. Bilindiği gibi bu model aslında Önyüklemeli-DQN konseptine göre tasarlanmıştır ve bu yüzden kafa olarak adlandırılan çoklu Q fonksiyonu çıktılarına sahiptir. Bu nedenle bu çıktı katmanı bu kafa sayısı kadar çıktı üretmektedir ve bu kafa sayısı 12 olarak ayarlanmıştır.

Daha önceden de bahsedildiği üzere  $\epsilon$ -açgözlü yaklaşımı kullanılarak bu modeller eğitilmiştir. Bu  $\epsilon$  değeri başlangıçta 1.0 olarak ayarlanıp 0.01 olana kadar bölüm sayısına göre belirli oranda azaltılarak uygulanmıştır. Optimize edici olarak Uyarlanabilir Moment Tahmini (Adaptive Moment Estimation, ADAM, [44]) kullanılmıştır ve her 200 bölümde eğitilen modeller kaydedilmiştir. Bunun dışında modelde izin verilen halka sayıları 5, 6 ve 7 olarak ayarlanmıştır. Geriye kalan diğer parametreler deneye göre değişmektedir ve deneylerin daha detaylı açıklandığı alt bölümlerde ise bu ayarlardan bahsedilmiştir.

### 5.1.2 Çizge tabanlı modeller

Her iki çizge sinir ağı mimarisine sahip model de bilindiği üzere aynı toksisite tahmini görevi için eğitilmiştir. Bu nedenle buradaki parametreler daha önceden bu görev için belirlenmiştir ve bu ayarlar değiştirilmemiştir.

Çizge Evrimsel Ağı (GCN) modelinde, molekülün çizge temsillerinin girdi olarak verildiği bir giriş katmanı ardından toplanma farklılıklarına (evrimsel mimariye) göre ayarlanan iki tane toplama katmanından oluşmaktadır. Bu toplama katmanları daha sonra okuma katmanından geçirilerek tam bağlı ileri beslemeli bir derin yapay sinir ağına aktarılıp toksisite tahmini için çıktılar alınmaktadır. Çizge yapılarının girdi olarak verildiği giriş katmanı 74 düğümden oluşmaktadır. Buradaki çıktının girdi olarak verildiği toplama katmanları ise 128 düğümden oluşmaktadır. Yine aynı şekilde toplama

katmanı da 128 düğüm sayısına sahiptir. Bu okuma katmanındaki 128 düğümünden alınan çıktılar molekül temsili olarak önerilen modelde kullanılmıştır.

Çizge Dikkat Ağı (GAT), ayarları bakımından GCN modelinden çok farklı değildir. Burada da çizge temsillerinin girdi olarak verildiği bir girdi katmanı ardından toplanma mimarisine göre ayarlanmış (dikkat mimarisi) iki adet toplama katmanı ve ardından bu çıktıların girdi olarak verildiği okuma katmanından oluşmaktadır. Daha sonra aynı şekilde okuma katmanından alınan çıktılar tam bağlı ileri beslemeli bir derin yapay sinir ağına girdi olarak verilip toksisite tahmini elde edilmektedir. Girdi katmanı GCN’de olduğu gibi 74 düğümünden oluşmaktadır. Farklı olarak GAT’da, okuma katmanlarından ilki 128 düğümünden oluşurken ikincisi 64 düğümünden oluşmaktadır. Ardından gelen okuma katmanı da yine 64 düğümünden oluşmaktadır. Okuma katmanındaki bu 64 düğümünden alınan çıktılar moleküler temsilleri oluşturmaktadır ve önerilen modelde bu çıktılar kullanılmıştır.

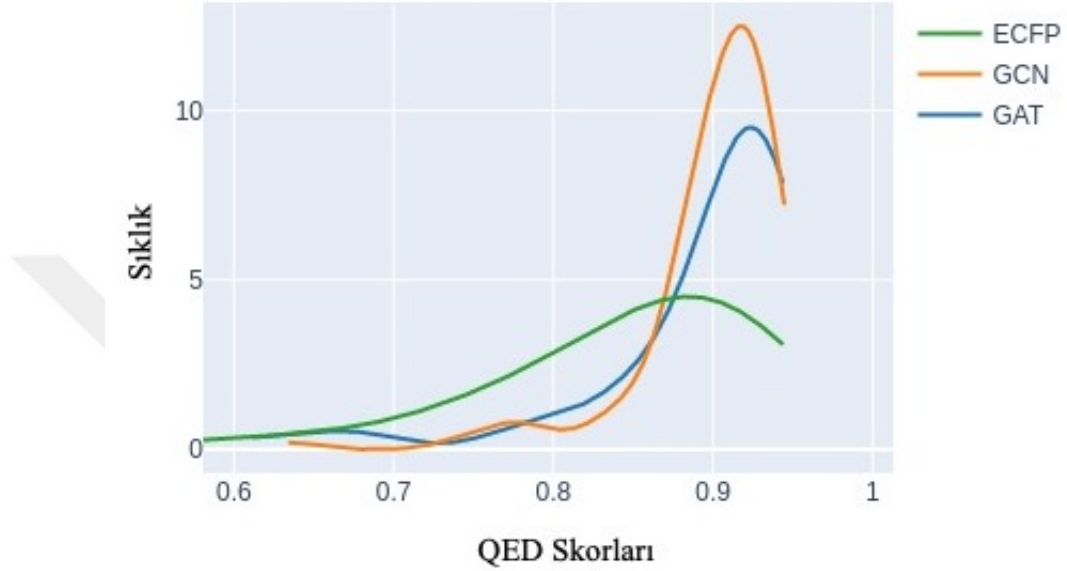
## 5.2 QED Optimizasyonu

Bu deney, farklı molekül temsillerini kullanarak QED puanını en üst düzeye çıkararak molekülü bulmayı ve bu sonuçlara dayalı olarak farklı molekül temsillerini kıyaslamayı amaçlar. Ödül fonksiyonu, molekülün QED değeri olarak ayarlanmıştır. [3]’te olduğu gibi, her bölüm için başlangıç durumu (molekülü) boş olarak ayarlanmıştır ve her bölüm için adım sayısı 40’tır. Toplam bölüm (iterasyon) sayısı ise 5000’dir. Atom eklenmesi için uygulanan eylemlerin öğelerinin kümesi {C, O, N} olarak ayarlanmıştır. Bunun yanında, eylem yapmama eylemi de ayarlanmıştır. Yani her adımda mutlaka bir eylem gerçekleştirmeyebilir. Ek olarak, moleküler halka aralarında bağ yapılacak eylemlere izin verilmeyecek şekilde ayarlanmıştır. Öğrenme oranı 0.0001, yığın boyutu (batch size) 128, tekrar arabellek boyutu (replay buffer size) 5000 ve indirim faktörü (discount factor) 0.9 olarak ayarlanmıştır. Bölüm 4’te detayları anlatılan modeller kullanarak ECFP, GCN ve GAT temsilleri için deneyler gerçekleştirilmiştir. Her temsil için kullanılan parametreler tamamen aynıdır. Nitekim tek fark, GAT, GCN ve ECFP temsillerinin boyutları farklı olduğu için ileri beslemeli sinir ağının giriş katmanıdır.

[3]’te gerçekleştirilen deneye benzer şekilde, eğitim aşamasının son 100 bölümünde ortaya çıkan (her bölümün nihai adımındaki) moleküller seçilmiştir ve en büyük QED değerine sahip ilk üç benzersiz molekül çizelge 5.1’de gösterilmiştir. Ek olarak, modelleri daha geniş bir açıdan bakarak kıyaslamak için şekil 5.1’de, son 100 bölümde üretilen moleküllerin QED değerlerinin (ödüllerin) dağılımı görselleştirilmiştir.

Çizelge 5.1: En iyi üç molekülün QED skorları.

Temsiller	QED Skorları		
	Birinci	İkinci	Üçüncü
ECFP	0.933	0.932	0.930
GCN	0.945	0.943	0.942
GAT	0.942	0.936	0.932



Şekil 5.1: Son 100 adımda üretilen moleküllerin QED skorlarının dağılımı.

### 5.3 Benzerlik Kısıtlaması ile Cezalı LogP Değerinin Optimizasyonu

[3]'teki bir diğer LogP optimizasyon deneyinde gösterildiği gibi, molekülleri yalnızca LogP değerine göre optimize etmenin gerçekçi olmayan uzun zincirli moleküller ürettiği gösterilmiştir. Bu sorunu çözmek için ilaç tasarımında en yaygın olarak kullanılan yöntemlerden biri, mevcut moleküle (ilaca) benzer ancak daha iyi aktiviteye sahip bir molekül bulmaya çalışmaktır. Üretilen moleküllerin mevcut moleküllere benzer olması gerektiğinden, gerçekçi olmayan uzun zincirler genellikle oluşmaz. Dolayısıyla bu deneyde bir molekülün, moleküllerin benzerliğini korurken, başlangıç molekülüne göre en yüksek cezalı LogP değerine ulaşması amaçlanmıştır. Bu nedenle, modelleri eğitirken iki metrik hesaplanması gerekiyordu. Bu metriklerden ilki, oluşturulan molekülün ilk moleküle olan benzerliğini ölçen bir benzerlik skorudur. Diğeri ise her adımda üretilen molekülün cezalı LogP değeridir. Cezalı LogP değeri (*PenLog*) denklem 5.1'de gösterilmiştir.

$$PenLogP(m) = LogP(m) - SA(m) - Cycle(m) \quad (5.1)$$

$LogP(m)$  burada  $m$  molekülünün LogP skorudur.  $SA(m)$ ,  $m$  molekülünün sentetik erişilebilirlik skorunun tahmini değeridir ve rdkit kütüphanesi kullanılarak hesaplanır ([45]).  $Cycle(m)$  ise  $m$  molekülünün en büyük halka değerini belirtir.

[3]'te yapılan benzerlik kısıtlaması ile cezalı LogP değerinin optimizasyonu deneyinin aksine burada benzerlik skoru, moleküllerin MACCS temsilleri arasındaki Tanimoto benzerliği (5.2) kullanılarak hesaplanır.

$$T = \frac{N_c}{N_a + N_b - N_c} \quad (5.2)$$

Burada  $a$  ve  $b$  molekülleri için  $N_a$  ve  $N_b$ , temsil vektörlerinde 1 olan toplam bit sayısını belirtir.  $N_c$  ise temsil vektörlerinde ortak ve 1 olan toplam bit sayısını belirtir.

ECFP gösterimi kullanılarak benzerlik puanı hesaplandığında, molekülün vektörel gösterimi uzun olduğundan (2048 bit) benzerlik puanı küçük değişikliklerle bile çok fazla azabilmektedir. Bu nedenle benzerlik kısıdı ile hesaplanan ödül değeri bundan fazla etkilenebilir ve çok düşük olabilmektedir. Böylece, penLogP değeri yeterince optimize edilememektedir. MACCS gösterimleri ise 166 ikili yapısal özelliğe sahip olduğundan bu sorun yaşanmamıştır. Rdkit kütüphanesinden Chem.MACCSkeys fonksiyonunu kullanarak MACCS temsilleri çıkarılmıştır. Şekil 5.2'de başlangıç molekülü olarak rastgele seçilmiş bir molekül kullanılmıştır ve 0.6'lık bir benzerlik kısıtlaması değeri ile eğitilen modellerin ilk beş adımının çıktıları, ECFP ve MACCS molekül temsilleri kullanılarak görselleştirilmiştir. Bu çıktılara bakıldığında benzerlik skoru değerlerinin değişiklikleri gözlemlenebilir. İlk beş adım olmasına rağmen, ECFP ile hesaplanan benzerlik değerleri MACCS ile hesaplanan benzerlik değerlerinden daha hızlı ve hatta 0,6'nın altına düşmektedir.

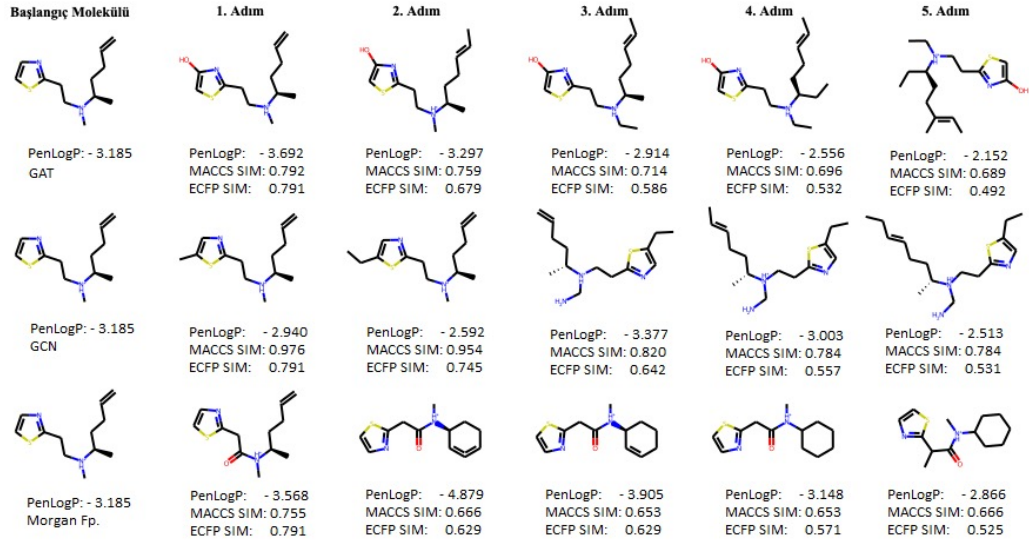
Ek olarak, Şekil 5.3'te, rastgele seçilmiş beş molekül üzerinde küçük değişiklikler yapıldığında Tanimoto benzerlik puanlarının sonuçları gösterilmiştir. Sonuçlardan görülebileceği gibi, özellikle küçük moleküllerde ECFP gösterimi kullanıldığında benzerlik puanı büyük ölçüde etkilenmiştir. Bu nedenle, benzerlik skorunun gerekli olduğu deneylerde MACCS temsili seçilmiştir.

Bir başka dikkat edilmesi gereken nokta, benzerlik skorunun aralığı  $[0, 1]$  iken, cezalı LogP değerinin aralığı  $(-\infty, +\infty)$ 'dur. Bu nedenle doğal olarak, bu değerleri daha dengeli yapmak için bir ağırlık katsayısına ihtiyacımız vardır. Bu yüzden, ödül değerinin fonksiyonu aşağıdaki gibi tanımlanmıştır.

$$R(s) = \begin{cases} PenLogP(m) - \lambda \times (\delta - SIM(m, m_0)), & \text{Eğer } SIM(m, m_0) \leq \delta \\ PenLogP(m), & \text{Aksi durumda} \end{cases}$$

[3]'te olduğu gibi burada da  $\lambda$ , cezalı LogP ve benzerlik puanını dengelemek için kullanılan bir değerdir ve 100 olarak ayarlanmıştır.  $\delta$  bu deneydeki benzerlik oranlarıdır





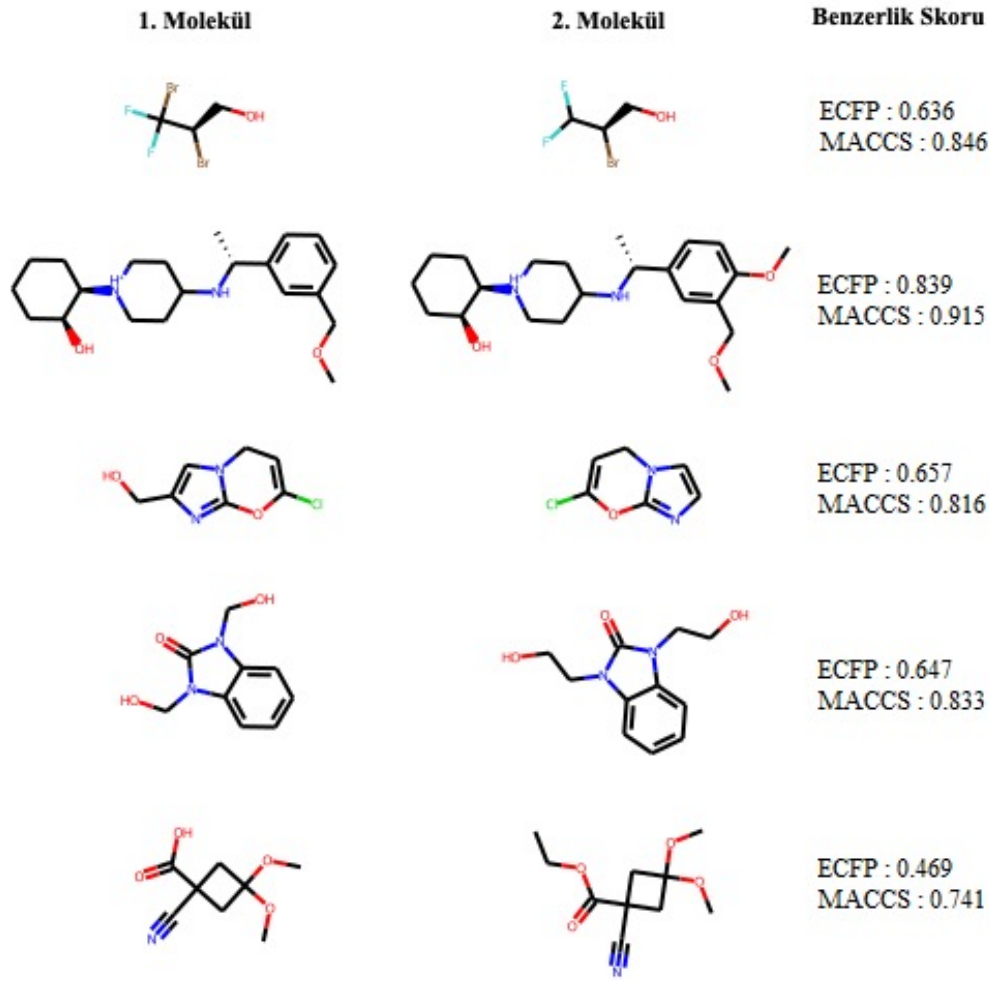
Şekil 5.2: 0,6 benzerlik kısıtlaması değeriyle ile cezalı LogP optimizasyonu için eğitilen modellerin ilk 5 adımında oluşan çıktılarının örnekleri (birinci sıra GAT ile, ikinci sıra GCN ile, üçüncü sıra ise ECFP ile eğitilmiştir ve SIM benzerlik skorunu göstermektedir).

ve 0.6, 0.4, 0.2, 0.0 değerleri ayarlanarak deneyler yapılmıştır.

Bu deneyde ([3]'te de aynı şekilde kullanılan) ZINC [46] veri tabanından seçilen en düşük cezalı LogP skoruna sahip 800 molekül [47] optimize edilmek için kullanılmıştır. Her bölüm için başlangıç noktası olarak bu 800 molekül arasından rastgele bir molekül seçilmiştir. Toplam bölüm sayısı 40000 olarak ayarlanmıştır ve her bölüm 20 adımdan oluşmaktadır. İlk deneyde de olduğu gibi atom ekleme eyleminin öğelerinin kümesi {C, O, N}'den oluşmaktadır. Yine aynı şekilde, moleküler halka aralarında bağ yapılacak eylemlere izin verilmemiştir ve her adımda boş eylem yapabilmektedir. Öğrenme oranı 0.0001, yığın boyutu 256, tekrar arabellek boyutu 10000 ve indirim faktörü 0.9 olarak ayarlanmıştır. Üç farklı molekül temsili için toplam dört benzerlik oranı kullanarak sonuçlar elde edilmiştir. [3] ile aynı şekilde, eğitim işlemi tamamlandıktan sonra, bu eğitilen modelleri kullanarak 800 molekülün her biri başlangıç noktası olacak şekilde sırasıyla tek bölüm (her başlangıç molekülü için 20 adım) çalıştırılmıştır. Ardından, üretilen bu yeni moleküllerin cezalı LogP puanları yeniden hesaplanmıştır ve 800 molekülün tümünün ortalama artışları çizelge 5.2'de bildirilmiştir.

#### 5.4 Benzerlik Kısıtlaması ile QED Optimizasyonu

Bu deneydeki amaç da benzerlik kısıtlaması ile cezalı LogP değerinin optimizasyonu deneyine benzerdir. Cezalı LogP değeri yerine burada bir molekülün QED puanını bir başlangıç molekülüne benzer tutarken en üst düzeye çıkarmak amaçlanmıştır. İkinci deneyden farklı olarak ödül fonksiyonunda, benzerlik skoru ve QED skoru aynı aralıkta



Şekil 5.3: ECFP ve MACCS temsillerinin benzerlik skorlarına ilişkin kıyası.

Çizelge 5.2: Ortalama cezalı LogP artışı.

$\delta$	Cezalı LogP Artışı.		
	ECFP	GCN	GAT
0.0	$6.64 \pm 1.38$	$8.31 \pm 0.83$	$8.48 \pm 0.79$
0.2	$5.97 \pm 2.00$	$8.19 \pm 0.86$	$8.23 \pm 0.80$
0.4	$5.75 \pm 1.98$	$6.86 \pm 1.13$	$7.55 \pm 1.13$
0.6	$4.57 \pm 2.34$	$4.94 \pm 2.26$	$6.78 \pm 2.26$

([0, 1]) değerler alabileceğinden, bu değerlerin toplamı 1 olan iki farklı katsayı (ağırlık çarpanı) ile çarpılarak aşağıdaki denklem formüle edilmiştir.

$$R(s) = w \times SIM(m, m_0) + (1 - w) \times QED(m) \quad (5.3)$$

$m_0$  hedef molekülünü belirtir.  $w$  ağırlık katsayılarıdır ve 0.0, 0.2, 0.4, 0.6, 0.8, 1.0 değerleri ayarlanarak deneyler yapılmıştır.

Bu deneyde, modele girdi olarak iki molekül verilmiştir. Bunlardan ilki başlangıç molekülü, diğeri ise hedef (benzerlik için) molekülüdür. Bu optimizasyonda yine [3]'teki gibi her iki girdiye de aynı molekül verilerek deneyler yapılmıştır. Toplamda altı ağırlık katsayısı ayarlanarak deneyler yapılmıştır. Her ağırlık katsayısı için model 3000 bölüm boyunca eğitilmiştir ve eğitim aşamasının son 100 bölümü içinden nihai adımda üretilen en büyük ödül değerine sahip molekül seçilmiştir. Her bölümdeki adım boyutu 40 olarak belirlenmiş olup, atom ekleme eylem kümesi ve indirim faktörü birinci ve ikinci deneylerdeki ile aynıdır. Geri kalan ayarlar ise QED optimizasyonu deneyi ile aynıdır. Çizelge 5.3'teki sonuçlar, girdi için (başlangıç ve hedef molekül olarak) 'CCN1c2ccccc2Cc3c(O)ncnc13' (ChEMBL7404) molekülü verilerek üç farklı molekül temsili ile ulaşılan en iyi QED ve benzerlik değerlerinin sonuçlarını gösterir. QED sütunu QED skorlarını belirtirken, SIM sütunu benzerlik skorlarını belirtir. Ayrıca, tüm ağırlık katsayıları için son 100 bölümde üretilen moleküllerin toplam ödül değerleri birleştirilip şekil 5.4'te görülebileceği üzere dağılımı görselleştirilmiştir.

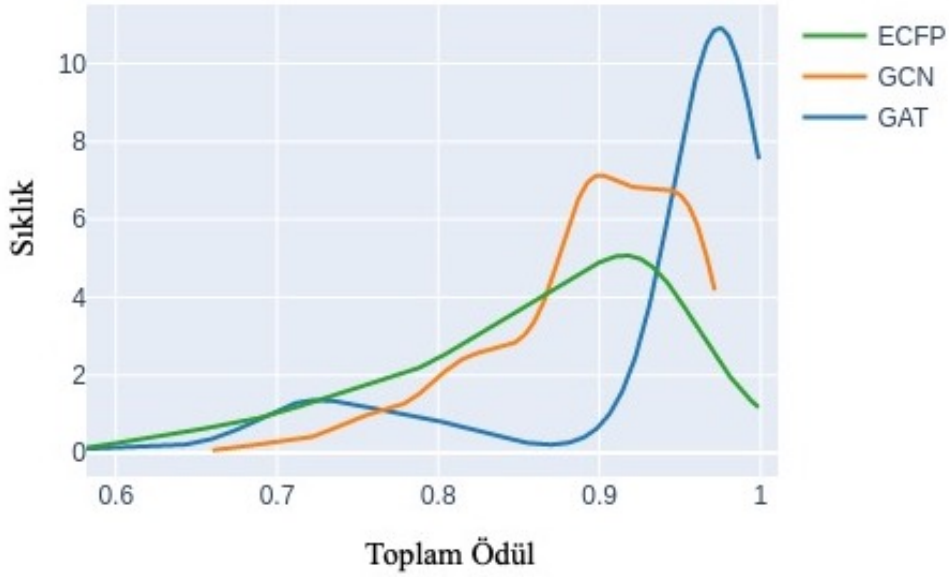
### 5.5 Hedef Proteine Kenetlenme Skorunun Optimize Edilmesi

Bu deneyde belirli hedef protein alıcılarına en iyi kenetlenen moleküler yapıların bulunması amaçlanmıştır. Ancak, ikinci deneye benzer şekilde yalnızca bu kenetlenme skorunu kullanarak molekül üretmek, kenetlenme skorlarının yüksek moleküler ağırlığa ve lipofilikliğine bağlı olduğu için yanlış sonuçlar verebilmektedir. Bu yüzden, ilaca benzemeyen büyük ve hidrofobik moleküllerin üretilmesine yol açılabilir [41]. Bu sorunun önüne geçmek için yukarıda yapılan deneylere benzer şekilde QED skoru ile birlikte kenetlenme skorunu belirli oranda birleştiren bir ödül fonksiyonu kullanılmıştır. Diğer deneylere benzer şekilde bu deneyde de elde edilen üç farklı moleküler temsil (GAT, GCN, ECFP) kullanılarak modeller eğitilmiştir ve birbirleri ile karşılaştırılmıştır. [41]'da uygulanan deneylere benzer şekilde, uygulanabilirliği açısından farklı zorluklara sahip üç farklı deney ve üç farklı protein alıcısına (F2, PPAR, JAK2) göre moleküllerin kenetlenebilirliği optimize edilmiştir. Seçilen hedeflerin detayları ve alınan sonuçlar bölüm 5.5.1, 5.5.2 ve 5.5.3'te aktarılmıştır. AutoDock Vina [48] [41]'da, yüksek kalitede kenetlenme ve serbest enerji bağlama skorlarının tahmini değerlerini hesaplamak için kullanılmıştır. Bu nedenle bu çalışmada da aynı şekilde kenetlenme skorları hesaplanmıştır.

Başlangıç molekülü her bölüm için boş olarak ayarlanmıştır ve toplam iterasyon sayısı 5000'dir. Atom ekleme eylemlerinin kümesi {C, O, N} olarak ayarlanmıştır ve yine diğer deneylerde olduğu gibi halka aralarında bağ yapılacak eylemlere izin verilmemiştir. Yine aynı şekilde her bölümde atom ekleme, bağ ekleme veya çıkarma eylemlerinden

Çizelge 5.3: En iyi ödül değerine sahip moleküllerin QED ve benzerlik skorları.

$w$	ECFP		GCN		GAT	
	QED	SIM	QED	SIM	QED	SIM
0.0	0.945	0.823	0.946	0.444	0.947	0.733
0.2	0.935	0.913	0.944	0.788	0.947	0.837
0.4	0.930	0.933	0.946	0.851	0.945	0.854
0.6	0.896	0.976	0.833	0.840	0.915	0.909
0.8	0.896	0.976	0.896	0.976	0.833	0.976
1.0	0.847	0.976	0.820	0.953	0.833	0.976



Şekil 5.4: Son 100 adımda üretilen moleküllerin ödül değerlerinin dağılımı.

biri yapılmayabilir. Öğrenme oranı 0.0001, yığın boyutu 128, tekrar arabellek boyutu 5000, indirim faktörü değeri ise 0.9 olarak ayarlanmıştır.

### 5.5.1 F2 (protease) kenetlenme

F2 (protease) deneyi, özetle hedef F2 alıcısına en iyi kenetlenen molekülü üretmek için yapılmaktadır. [41]'da da bahsedildiği üzere tek bir hedeften alınan kenetlenme skoruna göre optimize edildiği için zorluk derecesi kolaydır. Denklem 5.4'te F2 ile QED değeri birleştirilerek oluşturulan ödül fonksiyonu verilmiştir.

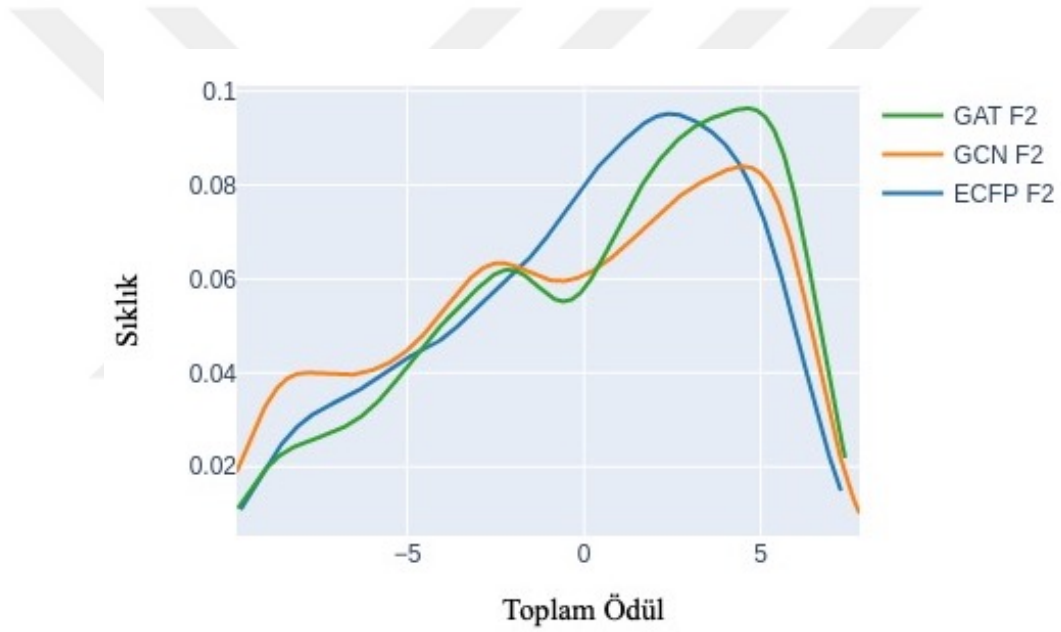
$$R_{f_2}(s) = F2(m) + 10(1 - QED(m)) \quad (5.4)$$

Burada  $F2(m)$ ,  $m$  molekülünün F2 alıcısına kenetlenme skorunu veren fonksiyonu

belirtir. Bu ödül fonksiyonu optimize edilerek üç farklı molekül temsili ile eğitilen modeller kıyaslanmıştır. Bu üç modeli kıyaslamak için ilk deneye benzer şekilde çizelge 5.4'te, eğitim aşamasının son 100 bölümünün nihai adımında üretilen en iyi üç molekülün ödül değerleri gösterilmiştir. Buna ek olarak yine son 100 adımda üretilen moleküllerin ödül değerlerinin dağılımı şekil 5.5'te verilmiştir.

Çizelge 5.4: En iyi ödül değerine sahip üç molekül (F2).

Temsiller	F2 Ödül Değerleri		
	Birinci	İkinci	Üçüncü
ECFP	7.300	7.289	7.273
GCN	7.828	7.489	7.329
GAT	7.423	7.400	7.386



Şekil 5.5: Üretilen moleküllerin ödül değerlerinin dağılımı (F2 için).

### 5.5.2 PPAR (nuclear receptors) kenetlenme

Bu deney, üç PPAR (PPARA, PPARD, PPARG) protein alıcısına güçlü bağlanma gerektirir ve bu üç PPAR kenetlenme skorları birbiri ile pozitif ilişkilidir. Bu nedenle orta zorlukta bir deneydir [41]. Metabolik sendroma karşı tedaviler olarak "rastlantısal (promiscuous)" pan-PPAR agonistleri sık araştırılan konulardan biridir. Hem bu nedenle hem de uygulanabilirliği düşünülünce, [41]'da kullanıldığı gibi burada da geliştirilen modeli değerlendirmek için tercih edilmiştir. Denklem 5.5'te PPAR ile QED değeri birleştirilerek oluşturulan ödül fonksiyonu verilmiştir. Ayrıca modellerin daha genel

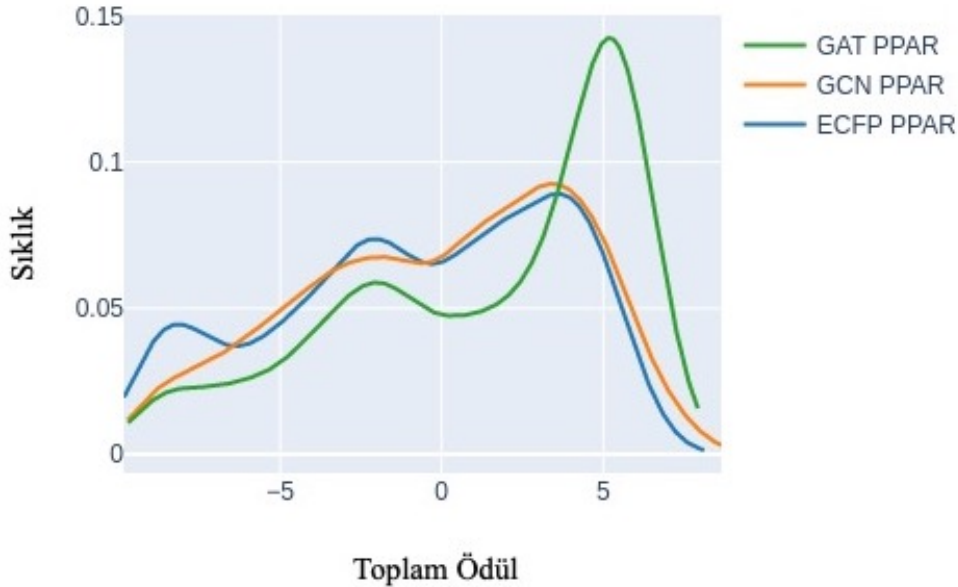
sonuçlarını kıyaslamak için yine son 100 adımda üretilen moleküllerin ödül değerlerinin dağılımı şekil 5.6'da çıkarılmıştır.

$$R_{PPAR}(s) = \max(PPARA(m), PPARD(m), PPARG(m)) + 10(1 - QED(m)) \quad (5.5)$$

Burada  $PPARA(m)$ ,  $PPARD(m)$  ve  $PPARG(m)$ ,  $m$  molekülünün ilgili protein alıcısına kenetlenme skorunu veren fonksiyonu belirtir. Pozitif ilişkili olduğu için en yüksek kenetlenme skoru ödül fonksiyonunda kullanılır. Bu ödül fonksiyonu optimize edilerek üç farklı molekül temsili ile eğitilen modeller kıyaslanmıştır. Bu üç modeli kıyaslamak için ilk deneye benzer şekilde çizelge 5.5'te, eğitim aşamasının son 100 bölümünün nihai adımında üretilen en iyi üç molekülün ödül değerleri gösterilmiştir.

Çizelge 5.5: En iyi ödül değerine sahip üç molekül (PPAR).

Temsiller	PPAR Ödül Değerleri		
	Birinci	İkinci	Üçüncü
ECFP	8.161	7.416	7.145
GCN	8.690	8.073	8.056
GAT	7.967	7.726	7.531



Şekil 5.6: Üretilen moleküllerin ödül değerlerinin dağılımı (PPAR için).

### 5.5.3 JAK2 (kinase) kenetlenme

Bu deney, JAK2 alıcısına güçlü bağlanma, LCK'ye ise zayıf bağlanma gerektirir. Buradaki zorluk, her ikisi de kinaz (kinase) olduğu için kenetlenme skorlarının pozitif korelasyona sahip olmasıdır. Kinazlar kanser ve hücre sinyalizasyonu rolleri sebebiyle tercih edilmektedir. Ancak seçiciliğe ulaşmak zordur. Ayrıca, hedef dışı etkiler ve toksisite yaygın olarak gözlenmektedir [41]. Denklem 5.6'da JAK2 ve LCK ile QED değeri birleştirilerek oluşturulan ödül fonksiyonu verilmiştir.

$$R_{JAK2}(s) = JAK2(m) - \min(LCK(m), -8.1) + 10(1 - QED(m)) \quad (5.6)$$

Burada  $JAK2(m)$  ve  $LCK(m)$ ,  $m$  molekülünün ilgili alıcısına kenetlenme skorunu veren fonksiyonu belirtir. LCK'ya zayıf kenetlenme gerektiği için dayanak noktası olarak medyanı (-8.1) kullanılmıştır. Yukarıdaki deneylerde olduğu gibi üç farklı molekül temsili ile eğitilen modeller, bu ödül fonksiyonunu optimize etmeyi amaçlar. Çizelge 5.6'da, eğitim aşamasının son 100 bölümünde üretilen en iyi üç molekülün ödül değerleri verilmiştir. Ayrıca yine son 100 bölümde üretilen moleküllerin ödül dağılımı şekil 5.7'de sunulmuştur.

Çizelge 5.6: En iyi ödül değerine sahip üç molekül (JAK2).

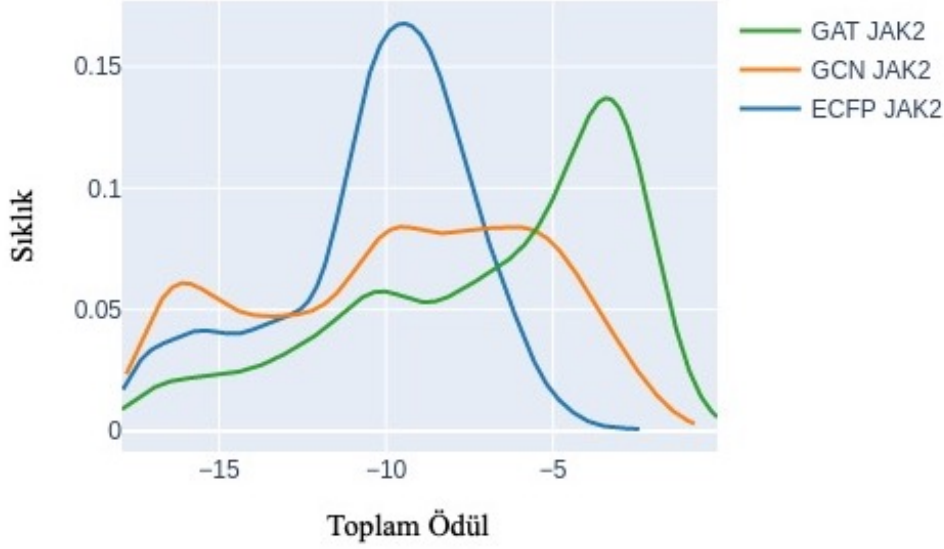
Temsiller	JAK2 Ödül Değerleri		
	Birinci	İkinci	Üçüncü
ECFP	-2.381	-2.465	-2.821
GCN	-0.727	-0.772	-1.545
GAT	-0.040	-0.420	-0.557

### 5.6 Öğrenilmiş Q Fonksiyonu ile Molekül Üretimi

Bu deneyde protein alıcılarına kenetlenme deneylerinde eğitilen ve alınan sonuçlara göre en iyi çalışan model kullanılarak moleküller üretilmiştir. Eğitilen modelin aksine sıfırdan moleküller tasarlanmamıştır, belirli rastgele başlangıç noktaları optimize edilerek alınan sonuçlar değerlendirilmiştir. Bu deneyin asıl amacı, gerçek dünya problemlerinde eğitilen modellerin bir şekilde uygulanmasını ve sonuçlarının nasıl olduğunu veya olabileceğini gözlemlemektir.

Yapılan deneyleri kıyaslamak için bildiğimiz kadarıyla bir temel yöntem olmadığından ötürü, takviyeli öğrenme tekniklerini değerlendirmek için literatürde uygulanan nispeten basit bir yöntem tercih edilmiştir. Bu yöntemde eğitilen modelden alınan ve tahmin edilen en iyi eylemler kullanılarak üretilen moleküllerin ödül değerleri, tamamen rastgele eylemler kullanılarak üretilen moleküllerin ödül değerleri ile kıyaslanır. Bu





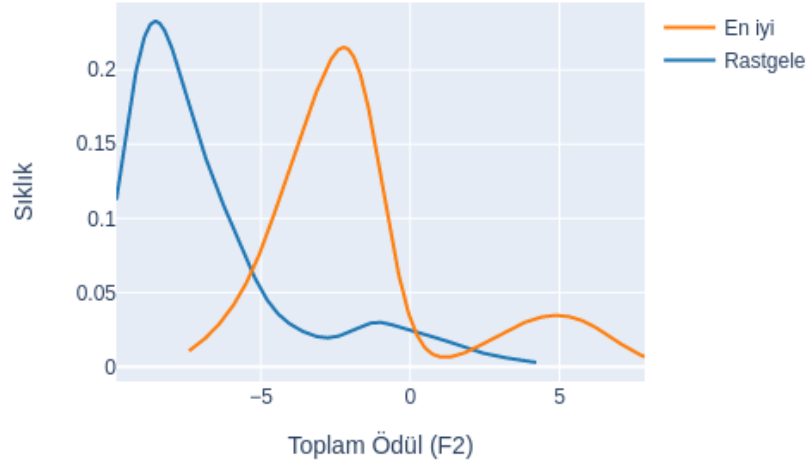
Şekil 5.7: Üretilen moleküllerin ödül değerlerinin dağılımı (JAK2 için).

şekilde en azından tamamen rastgele seçilen eylemlere göre daha iyi sonuçlar alınıp alınmadığı ölçülür.

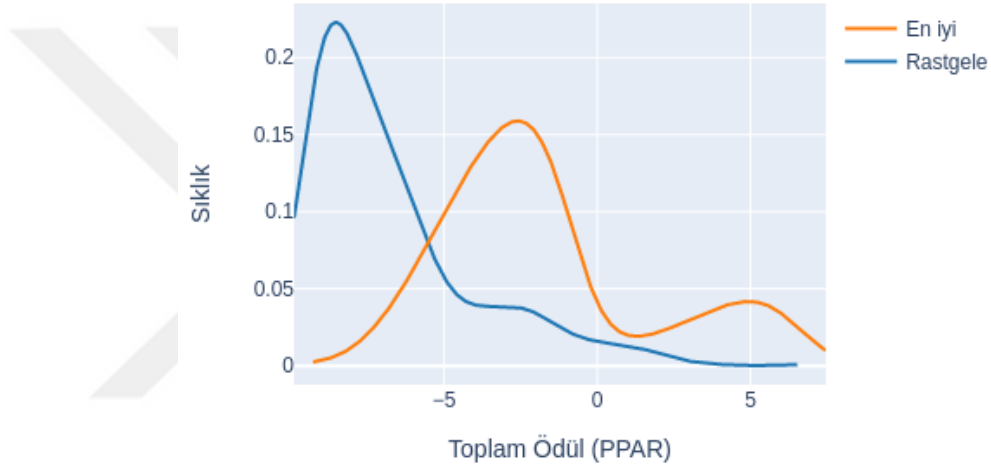
[41]'da yapılan deneylerde ZINC veri tabanından seçilen rastgele moleküller başlangıç noktaları kullanılarak optimize edilmiştir. Ancak buradaki sonuçlar ile kıyaslamak için bu başlangıç noktalarının bilinmesi gerekmektedir ve bu bilgi [41]'da bilindiği kadarıyla paylaşılmamıştır. Bunun için bu tez çalışmasında, benzerlik kısıtlaması ile cezalı LogP değerinin optimizasyonu deneyinde kullanılan ZINC veri tabanından seçilen 800 molekül, burada rastgele başlangıç noktaları olarak kullanılmıştır. Bahsedildiği üzere, kenetlenme deneylerinde değerlendirmeler sonucu elde edilen en iyi modeli (GAT) kullanarak, üç farklı hedef protein alıcısı (F2, PPAR, JAK2) için 800 başlangıç noktasını girdi olarak vererek moleküller üretilmiştir.

Her bir deneyde başlangıç noktası olarak verilen molekül üzerinde, öğrenilen en iyi modelin Q fonksiyonu kullanılarak 20 adımlık modifikasyon yapılmıştır. Bu sonuçları kıyaslamak için de her başlangıç noktası için 20 adımlık rastgele modifikasyonlar sonucu üretilen moleküller kullanılmıştır. Her iki yöntemle üretilen moleküllerin toplam ödül değerlerinin dağılımları diğer deneylere benzer şekilde çıkarılmıştır. Şekil 5.8'de, F2 için eğitilen modelden alınan sonuçların, toplam ödül değerlerinin dağılımı verilmiştir. Şekil 5.9'da, PPAR protein alıcıları için eğitilen modelden alınan sonuçların, toplam ödül değerlerinin dağılımı verilmiştir. Son olarak, şekil 5.10'da, JAK2 protein alıcısı için eğitilen modelden alınan sonuçların toplam ödül değerlerinin dağılımı verilmiştir.

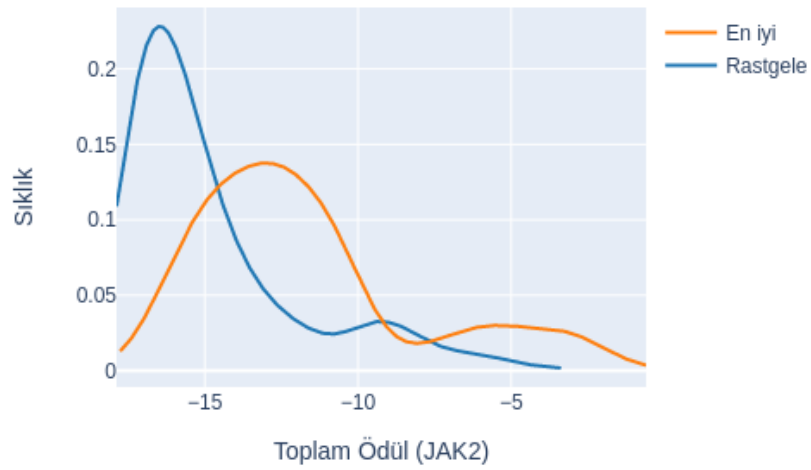




Şekil 5.8: Üretilen moleküllerin toplam ödül dağılımı (F2 için).



Şekil 5.9: Üretilen moleküllerin toplam ödül dağılımı (PPAR için).



Şekil 5.10: Üretilen moleküllerin toplam ödül dağılımı (JAK2 için).

## 6. SONUÇ

### 6.1 Değerlendirmeler ve Tartışma

Bu bölümde yapılan deneyler özetlenip, elde edilen sonuçlar tartışılmıştır. İlk deneyde, çizelge 5.1’de QED skoru ödül değeri olarak kullanıldığında, eğitim aşamasının son 100 bölümde üretilen moleküllerin sonuçları listelenmiştir. Görüldüğü gibi grafik tabanlı önceden eğitilmiş modellerden (GCN, GAT) alınan molekül temsilleri ile eğitilen model kullanılarak üretilen moleküllerin QED skorları, Morgan Parmak İzi temsilleri ile eğitilen model kullanılarak üretilen moleküllere göre daha yüksektir. Buna ek olarak şekil 5.1’de, son 100 adımda üretilen moleküllerin QED skorlarının dağılımı incelendiğinde, grafik tabanlı temsiller ile eğitilen modeller ve Morgan Parmak İzi temsilleri ile eğitilen model arasındaki fark daha iyi gözlenmektedir. Bu şekil, üretilen moleküllerin sıklığı ödül değeri arttıkça (QED skoru), çizge tabanlı temsillerde de önemli miktarda arttığını göstermektedir. Bu sonuçlara bakarak, tek amaçlı bir hedef olan QED optimizasyonu için grafik tabanlı temsillerle eğitilen modeller kullanılarak sıfırdan üretilen moleküllerin, deneysel anlamda temel yöntemle göre daha iyi bir noktaya ulaştığı söylenebilir.

İlk deneyden farklı olarak ikinci deneyde, birden fazla moleküler özellik kullanılarak ödül fonksiyonu optimize edilmiştir. Bölüm 5.3’te bahsedildiği üzere, molekülleri sadece tek bir özelliğe göre optimize etmek istenmeyen sonuçlar verebilmektedir. Burada da yalnızca cezalı LogP değerine göre optimize etmenin ilaca benzemeyen uzun zincirli moleküler yapıların oluşabileceğinden bahsedilmiştir. Bunun önünce geçmek için cezalı LogP değerine ek olarak, belirli oranda başlangıç molekülüne benzemesi amaçlanmıştır. Bunun için de belirli katsayılarla benzerlik kısıtları ödül fonksiyonunda ayarlanmıştır. Model eğitildikten sonra 800 rastgele başlangıç molekülü için cezalı LogP artışları çizelge 5.1’de verilmiştir. Bu sonuçlara bakıldığında, çizge tabanlı modellerden alınan molekül temsilleri ile eğitilmiş modellerin, temel yöntemle eğitilen modelden deneysel olarak çok daha başarılı olduğu söylenebilir. Özellikle benzerlik kısıtı oranının yüksek olduğu (0.6 ve 0.4) durumlarında, ortalama cezalı LogP iyileştirmelerindeki bu fark gerçekten önemlidir. Çünkü bu kısıdın yüksek olduğu şartların sağlanması, başlangıç molekülünden çok uzaklaşmadan daha iyi cezalı LogP skoru alındığını gösterir ve aslında amaçlanan da budur. Ayrıca, bu çalışmadaki ana motivasyonlardan biri de LogP ve toksisite arasındaki ilişkiyi kullanmaktır. Bu nedenle, önceden moleküler toksisiteyi tahmin etmek için eğitilmiş çizge tabanlı modellerin kullanılması tercih edilmiştir. Bu sonuçlar ayrıca, LogP ile toksisite arasındaki korelasyonun cezalı LogP değerinin gelişimi üzerinde olumlu bir etkisi olduğunu da göstermektedir.

Üçüncü deneyde de belirli bir benzerlik kısıtlaması ile QED değerinin optimizasyonu amaçlanmıştır. İlk deneye benzer şekilde eğitim aşamasının son 100 adımında üretilen moleküller arasından en yüksek ödül değerine ulaşan moleküllerin sonuçları çizelge 5.3'te gösterilmiştir. Bu sonuçlara bakıldığında hangi modelin daha iyi sonuçlar verdiği net değildir. Buna karşılık, şekil 5.4'te, son 100 adımda üretilen moleküllerin dağılımı çıkarılmıştır. Buradaki deneyde bilindiği üzere farklı ağırlıklardaki benzerlik kısıtlaması ile toplamda altı farklı ödül fonksiyonu kullanılmıştır. Bu nedenle, dağılımı çıkarırken buradaki sonuçların tümü birleştirilmiştir. Dağılıma bakıldığında GAT modeli ile diğer iki modele göre daha sık bir şekilde yüksek ödül değerlerine sahip molekül üretildiğini söyleyebiliriz.

Uygulanabilirliği ve zorluk seviyeleri olarak birbirinden farklı üç hedef protein alıcısına en iyi kenetlenen molekülü üretmek için eğitilen model, bu çalışmadaki dördüncü deneyi oluşturmaktadır. Daha önceden de bahsedildiği üzere bu deneyler, [41]'da ZINC veri kümesinden rastgele seçilen başlangıç molekülleri optimize edilerek yapılmıştır. Ancak [41]'da yapılan deneylerdeki başlangıç noktaları paylaşılmamıştır. Bu açıdan buradaki sonuçlar ile doğrudan bir kıyaslama yapılamadığı için diğer üç deneydeki gibi temel yöntem olarak Morgan Parmak İzi temsilleri ile eğitilen model kullanılmıştır. Temel yöntemle kıyaslamak için ilk deneyde olduğu gibi herhangi bir veri kümesi kullanılmadan sıfırdan üretilen moleküllerin skorları kıyaslanmıştır. Son 100 adımda üretilen en yüksek skora sahip üç molekülün listelendiği çizelgelere (çizelge 5.4 ve çizelge 5.5) bakıldığında, F2 ve PPAR protein alıcıları için kayda değer bir fark yoktur. Ancak son 100 adımda üretilen moleküllerin toplam ödül değerlerinin dağılımına bakıldığında (şekil 5.5 ve şekil 5.6), yine çizge tabanlı modellerden alınan sonuçların Morgan Parmak İzi ile alınan sonuçlardan çok daha iyi olduğu göze çarpmaktadır. Öte yandan, JAK2 protein alıcısı için yapılan deneylere bakıldığında hem üretilen en yüksek ödül değerine sahip üç molekülün değeri (çizelge 5.6) hem de üretilen moleküllerin ödül dağılımı (çizelge 5.7) açısından, çizge tabanlı modeller ile alınan sonuçlar çok daha iyidir. Daha önceden de bahsedildiği üzere, JAK2'da yapısı gereği toksisite yaygın olarak gözlenmektedir. Burada kullanılan çizge tabanlı modeller, toksisite tahmini için önceden eğitilen modeller olduğu için böyle bir fark olduğu söylenebilir.

Son deneyde ise protein alıcılarına kenetlenme deneyinde eğitilen ve en iyi sonuçların alındığı GAT modelinin öğrenilmiş Q fonksiyonu kullanılarak, rastgele seçilen 800 başlangıç noktası için deneyler yapılmıştır. Önerilen modeller ile belirli başlangıç noktalarına özel bir eğitim yapıldığı zaman ortaya çıkan sonuçların umut vadeden olduğu yapılan deneyler ile gösterilmiştir. Ancak, gerçek dünya uygulamalarında bu eğitilen modellerin hızlı bir şekilde kullanılması için başka ortamlarda veya başka başlangıç noktaları verildiğinde de başarılı olması istenir. Bu deneyde ise farklı başlangıç noktalarından üretilen moleküller beklendiği gibi yukarıdaki deneyler kadar iyi sonuçlar

vermemiştir. Fakat, şekil 5.8, şekil 5.9 ve şekil 5.10’da gösterildiği üzere, en iyi sonuçların alındığı öğrenilmiş Q fonksiyonu kullanılarak üretilen moleküllerin, burada temel yöntem olarak kullanılan ve tamamen rastgele bir şekilde değişiklik yaparak üretilen moleküllerden çok daha başarılı olduğu göze çarpmaktadır.

Daha önce de belirtildiği gibi, GAT ve GCN arasındaki temel fark, düğümün tek atlamalı komşusundan (one-hop neighborhood) gelen bilgilerin toplanma şeklidir. GAT, yapıdan bağımsız normalleştirme kullandığından, teorik olarak genelleme yeteneği GCN’lerden daha iyi olabilir. Buna paralel olarak çoğu deneyde öne çıkan bir diğer önemli nokta, GAT mimarisi ve çıkarılan temsiller kullanılarak eğitilen modelin özellikle toksisitenin dolaylı yoldan ilgili olduğu deneylerde GCN mimarisine göre çok daha iyi sonuçlar elde etmesidir.

Özetle, tasarlanan modeli değerlendirmek için özelliklerine ve amaçlarına göre beş farklı deney yapılmıştır. Genel olarak çizge tabanlı temsil kullanılarak eğitilen modellerin deneysel anlamda temel yöntemlere göre çok daha başarılı olduğu söylenebilir. Bunun dışında özellikle toksisite ile dolaylı yoldan ilişkili olan deneylerde önceden eğitilen GAT ve GCN modellerinin farkı dikkat çekmektedir.

## 6.2 Gelecek Çalışmalar

Takviyeli öğrenme modellerinin eğitim süreci yapıları gereği çok zaman alabilmektedir. Bu çalışmadaki takviyeli öğrenme modelinin eğitim sürecinde her bölüm için adım sayısı sınırlı olsa da bu işlem genel olarak uzun sürmektedir. Özellikle çizge tabanlı modellerden alınan molekül temsilleri kullanılarak eğitilen modellerde, her adımda bu çizge tabanlı yapay sinir ağlarının okuma katmanından o molekülün temsili alındığı için eğitim aşaması çok daha uzun sürmektedir. Takviyeli öğrenme algoritması gelecekte bu süreci hızlandırmak için değiştirilebilir. Özellikle son zamanlarda artan veriler ile makine öğrenme ve derin öğrenme algoritmalarının dağıtık bir şekilde eğitilmesi için geliştirilen yöntemler burada da takviyeli öğrenme için kullanılabilir. Takviyeli öğrenme modeli, dağıtık bir sistem üzerinde, birden çok ortam ve karar verici ajanı içeren bir tasarımda eğitilerek daha hızlı bir şekilde modelin yakınsanması sağlanabilir. Örneğin normalde eğitim aşamasında ideal ödül değerleri alması için gereken bölüm sayısı 3000 ise bu dağıtık sistemde daha kısa olabilir. Paralel bir şekilde bu eğitim sağlandığı için daha kısa sürede ideal ödüllerin toplandığı seviyeye ulaşılabilir.

Bunun dışında burada kullanılan çizge tabanlı modeller daha önceden de vurgulandığı üzere, hem daha hızlı bir şekilde eğitimin gerçekleştirilmesi hem de dolaylı yoldan toksisite ile bağlantısı olan moleküler özelliklerin incelenmesi için tercih edilmiştir. Bu modeller farklı bir veri kümesi kullanılarak moleküler toksisite tahmini için eğitilmiştir ve elde edilen sonuçlar toksisite ile ilgisi olmayan deneylerde bile dikkat çekicidir. Bu çizge tabanlı modeller de ortamdaki elde edilen verilere göre eğitilseydi çok daha iyi

sonular alınabilirdi. Ancak yukarıda da belirtildiđi gibi zaman aısından ok verimli olmadıđı iin yapılamadı. Eđer takviyeli ğrenme mimarisi sre aısından optimize edilirse veya dađıtık bir biimde eđitim uygulanabilirse, ileride bu izge tabanlı modeller bu mimariye uygun bir biimde ayarlanabilir ve performansları arttırılabilir.

Bunun dıřında, ne yazık ki bu alandaki alıřmaları kıyaslamak iin var olan metrikler yetersizdir. rneđin QED'nin sınır etkilerinden dolayı (0-1 aralıđı) iyi bir kıyaslama yapılamamaktadır ya da bu alıřmada da bahsedildiđi zere benzerlik hesaplarken kullanılan temsiller, sonuları ok etkilemektedir. Ayrıca, stn modeller geliřtirmek iin daha fazla "hesaplanabilir" (QED gibi) veya bir model, algoritma, program, vb. yardımıyla "tahmini" (kenetlenme skoru gibi) sayısal molekler zelliklere ihtiyacımız vardır. Neyse ki, bu alan hızla bymektedir ve yeni metriklerin veya sayısal molekler zelliklerin geliřtirilmesi kaınılmazdır. Buradaki geliřtirilen modelin dl fonksiyonu yeterli esnekliđe sahiptir. Bu sayede, gelecekteki modellerin dl fonksiyonu zahmet harcamadan deđiřtirilip, ayarlanabilir. Bylece, gerek dnya uygulamalarında (belirli hastalıklar iin ila retimi veya optimizasyonu gibi) burada tasarlanan modeller daha etkili bir biimde kullanılabilir.



## KAYNAKLAR

- [1] Sliwoski, G., Kothiwale, S., Meiler, J. Lowe, E. W. (2014). Computational methods in drug discovery. *Pharmacological reviews* 66.1, 334–395.
- [2] Dobson, C. M. (2004). Chemical space and biology. *Nature* 432.7019, Sf. 824–828.
- [3] Zhou, Z., Kearnes, S., Li, L., Zare, R. N. Riley, P. (2019). Optimization of molecules via deep reinforcement learning. *Scientific reports* 9.1, Sf. 1–10.
- [4] Kearnes, S., McCloskey, K., Berndl, M., Pande, V. Riley, P. (2016). Molecular graph convolutions: moving beyond fingerprints. *Journal of computer-aided molecular design* 30.8, Sf. 595–608.
- [5] Url-1 (2014). <https://tripod.nih.gov/tox21/challenge/data.jsp>. alındığı tarih: 29.04.2021.
- [6] Heipieper, H. J. Martinez, P. (2018). Toxicity of hydrocarbons to microorganisms. *Cellular Ecophysiology of Microbe: Hydrocarbon and Lipid Interactions*, Sf. 335.
- [7] Butler, K. T., Davies, D. W., Cartwright, H., Isayev, O. Walsh, A. (2018). Machine learning for molecular and materials science. *Nature* 559.7715, Sf. 547–555.
- [8] Archetti, F., Lanzeni, S., Messina, E. Vanneschi, L. (2007). Genetic programming for computational pharmacokinetics in drug discovery and development. *Genetic Programming and Evolvable Machines* 8.4, Sf. 413–432.
- [9] Elton, D. C., Boukouvalas, Z., Fuge, M. D. Chung, P. W. (2019). Deep learning for molecular design—a review of the state of the art. *Molecular Systems Design & Engineering* 4.4, Sf. 828–849.
- [10] Weininger, D. (1988). SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *Journal of chemical information and computer sciences* 28.1, Sf. 31–36.
- [11] Xu, Y., Lin, K., Wang, S., Wang, L., Cai, C., Song, C., Lai, L. Pei, J. (2019). Deep learning for molecular generation. *Future medicinal chemistry* 11.6, Sf. 567–597.

- [12] **Walters, W. P. Barzilay, R.** (2020). Applications of Deep Learning in Molecule Generation and Molecular Property Prediction. *Accounts of Chemical Research*, Sf. 3370–3388.
- [13] **Bjerrum, E. J. Threlfall, R.** (2017). Molecular generation with recurrent neural networks (RNNs). *arXiv preprint arXiv:1705.04612*.
- [14] **Kusner, M. J., Paige, B. Hernández-Lobato, J. M.** (2017). Grammar variational autoencoder. In: *International Conference on Machine Learning*. PMLR, Sf. 1945–1954.
- [15] **Blaschke, T., Olivecrona, M., Engkvist, O., Bajorath, J. Chen, H.** (2018). Application of generative autoencoder in de novo molecular design. *Molecular informatics* 37.1-2, Sf. 1700123.
- [16] **Knudde, N., Herten, J. van der, Dhaene, T. Couckuyt, I.** (2017). GPflowOpt: A Bayesian optimization library using TensorFlow. *arXiv preprint arXiv:1711.03845*.
- [17] **De Cao, N. Kipf, T.** (2018). MolGAN: An implicit generative model for small molecular graphs. *arXiv preprint arXiv:1805.11973*.
- [18] **Maziarka, Ł., Pocha, A., Kaczmarczyk, J., Rataj, K., Danel, T. Warchol, M.** (2020). Mol-CycleGAN: a generative model for molecular optimization. *Journal of Cheminformatics* 12.1, Sf. 1–18.
- [19] **Mercado, R., Rastemo, T., Lindelöf, E., Klambauer, G., Engkvist, O., Chen, H. Bjerrum, E. J.** (2021). Graph networks for molecular design. *Machine Learning: Science and Technology* 2.2, Sf. 025023.
- [20] **Pogány, P., Arad, N., Genway, S. Pickett, S. D.** (2018). De novo molecule design by translating from reduced graphs to SMILES. *Journal of chemical information and modeling* 59.3, Sf. 1136–1146.
- [21] **Neil, D., Segler, M., Guasch, L., Ahmed, M., Plumbley, D., Sellwood, M. Brown, N.** (2018). Exploring deep recurrent models with reinforcement learning for molecule design.
- [22] **Kraev, E.** (2018). Grammars and reinforcement learning for molecule optimization. *arXiv preprint arXiv:1811.11222*.
- [23] **Ståhl, N., Falkman, G., Karlsson, A., Mathiason, G. Bostrom, J.** (2019). Deep reinforcement learning for multiparameter optimization in de novo drug design. *Journal of chemical information and modeling* 59.7, Sf. 3166–3176.
- [24] **Jeon, W. Kim, D.** (2020). Autonomous molecule generation using reinforcement learning and docking to develop potential novel inhibitors. *Scientific reports* 10.1, Sf. 1–11.
- [25] **You, J., Liu, B., Ying, R., Pande, V. Leskovec, J.** (2018). Graph convolutional policy network for goal-directed molecular graph generation. *arXiv preprint arXiv:1806.02473*.



- [26] **Khemchandani, Y., O'Hagan, S., Samanta, S., Swainston, N., Roberts, T. J., Bollegala, D. Kell, D. B.** (2020). DeepGraphMolGen, a multi-objective, computational strategy for generating molecules with desirable properties: a graph convolution and reinforcement learning approach. *Journal of Cheminformatics* 12.1, Sf. 1–17.
- [27] **Rogers, D. Hahn, M.** (2010). Extended-connectivity fingerprints. *Journal of chemical information and modeling* 50.5, Sf. 742–754.
- [28] **Kaelbling, L. P., Littman, M. L. Moore, A. W.** (1996). Reinforcement learning: A survey. *Journal of artificial intelligence research* 4, Sf. 237–285.
- [29] **Pandian, B. J. Noel, M. M.** (2018). Control of a bioreactor using a new partially supervised reinforcement learning algorithm. *Journal of Process Control* 69, Sf. 16–29.
- [30] **Metropolis, N. Ulam, S.** (1949). The monte carlo method. *Journal of the American statistical association* 44.247, Sf. 335–341.
- [31] **Watkins, C. J. Dayan, P.** (1992). Q-learning. *Machine learning* 8.3, Sf. 279–292.
- [32] **Li, Y.** (2017). Deep reinforcement learning: An overview. *arXiv preprint arXiv:1701.07274*.
- [33] **Arulkumaran, K., Deisenroth, M. P., Brundage, M. Bharath, A. A.** (2017). Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine* 34.6, Sf. 26–38.
- [34] **Boyd, S., Boyd, S. P. Vandenberghe, L.** (2004). Convex optimization. Cambridge university press.
- [35] **Sutton, R. S., Barto, A. G., et al.** (1998). Introduction to reinforcement learning. Vol. 135. MIT press Cambridge.
- [36] **Osband, I., Blundell, C., Pritzel, A. Van Roy, B.** (2016). Deep exploration via bootstrapped DQN. *Advances in neural information processing systems* 29, Sf. 4026–4034.
- [37] **Sun, M., Zhao, S., Gilvary, C., Elemento, O., Zhou, J. Wang, F.** (2020). Graph convolutional networks for computational drug development and discovery. *Briefings in bioinformatics* 21.3, Sf. 919–935.
- [38] **Kipf, T. N. Welling, M.** (2016). Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.
- [39] **Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P. Bengio, Y.** (2017). Graph attention networks. *arXiv preprint arXiv:1710.10903*.
- [40] **Url-2** (2013). <https://www.rdkit.org/docs>. alındığı tarih: 12.01.2021.
- [41] **García-Ortegón, M., Simm, G. N., Tripp, A. J., Hernández-Lobato, J. M., Bender, A. Bacallado, S.** (2021). DOCKSTRING: easy molecular docking yields better benchmarks for ligand design. *arXiv preprint arXiv:2110.15486*.

- [42] Wang, M., Zheng, D., Ye, Z., Gan, Q., Li, M., Song, X., Zhou, J., Ma, C., Yu, L. Gai, Y. (2019). Deep graph library: A graph-centric, highly-performant package for graph neural networks. *arXiv preprint arXiv:1909.01315*.
- [43] Ioffe, S. Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: *International conference on machine learning*. PMLR, Sf. 448–456.
- [44] Kingma, D. P. Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- [45] Ertl, P. Schuffenhauer, A. (2009). Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *Journal of cheminformatics* 1.1, Sf. 1–11.
- [46] Irwin, J. J., Sterling, T., Mysinger, M. M., Bolstad, E. S. Coleman, R. G. (2012). ZINC: a free tool to discover chemistry for biology. *Journal of chemical information and modeling* 52.7, Sf. 1757–1768.
- [47] Url-3 (2019). [https://github.com/google-research/google-research/blob/master/mol\\_dqn/chemgraph/all\\_800\\_mols.json](https://github.com/google-research/google-research/blob/master/mol_dqn/chemgraph/all_800_mols.json). alındı̇ tarih: 04.02.2021.
- [48] Trott, O. Olson, A. J. (2010). AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *Journal of computational chemistry* 31.2, Sf. 455–461.